10.35662/unine-thesis-2862

DISSERTATION

DATA ASSIMILATION AND NON-GAUSSIAN PARAMETER INFERENCE FOR HYDROGEOLOGICAL MODELS

PhD committee (July 8th 2020):

Prof. Dr. Mario Schirmer

Prof. Dr. Philip Brunner

Dr. Carlo Albert

Prof. Dr. Olaf Cirpka

Maximilian Ramgraber

Centre d'hydrogéologie et de géothermie (CHYN)

Faculté des sciences

2020

unine
UNIVERSITÉ DE NEUCHÂTEL

Faculté des Sciences
Secrétariat-décanat de Faculté
Rue Emile-Argand 11
2000 Neuchâtel – Suisse
Tél : + 41 (0)32 718 21 00
E-mail : secretariat.sciences@unine.ch

IMPRIMATUR POUR THESE DE DOCTORAT

La Faculté des sciences de l'Université de Neuchâtel autorise l'impression de la présente thèse soutenue par

Monsieur Maximilian RAMGRABER

Titre:

"Data assimilation and non-Gaussian parameter inference for hydrogeological models"

sur le rapport des membres du jury composé comme suit:

- Prof. associé Mario Schirmer, directeur de thèse, Université de Neuchâtel, Suisse
- Prof. Philip Brunner, Université de Neuchâtel, Suisse
- Prof. Olaf Cirpka, Université de Tübingen, Allemagne
- Dr. Carlo Albert, Eawag, Dübendorf, Suisse

Neuchâtel, le 6 août 2020

Le Doyen, Prof. A. Bangerter

Summary

One of the fundamental challenges of hydrogeology is operating in an environment that remains mostly inaccessible and hence unobserved. Since geological coring is expensive, direct information about the subsurface is scarce. As a consequence, hydrogeologists must instead characterize the system through indirect information such as water table dynamics. To this end, Bayesian statistics provide an effective framework for converting indirect knowledge into estimates of subsurface properties. At the same time, it provides uncertainty estimates which reflect possible ambiguity. Unfortunately, it is generally difficult to capture all facets of this ambiguity. The assumption of Gaussianity, for example, permits very efficient numerical solutions, but represents only simple forms of uncertainty. However, when there could exist more than one distinct, functionally equivalent solution, such approaches prove too simplistic. Since such scenarios abound in hydrogeology, it is important to develop methods that could adequately capture the uncertainty in the subsurface. To this end, this dissertation explores the suitability of particle filters and variational methods for the inference of non-Gaussian hydrogeological parameter uncertainty.

L'un des défis fondamentaux de l'hydrogéologie est d'opérer dans un environnement qui reste pour la plupart inaccessible et donc non observé. Comme le carottage géologique est coûteux, les informations directes sur le sous-sol sont rares. En conséquence, les hydrogéologues doivent plutôt caractériser le système par des informations indirectes telles que la dynamique de la nappe phréatique. À cette fin, les statistiques bayésiennes fournissent un cadre efficace pour convertir les connaissances indirectes en estimations des propriétés de la subsurface. En même temps, elles fournissent des estimations d'incertitude qui

reflètent une possible ambiguïté. Malheureusement, il est généralement difficile de saisir toutes les aspects de cette ambiguïté. L'hypothèse de la gaussianité, par exemple, permet des solutions numériques très efficaces, mais ne représente que des formes simples d'incertitude. Cependant, lorsqu'il pourrait exister plus d'une solution distincte et fonctionnellement équivalente, de telles approches s'avèrent trop simplistes. Étant donné que de tels scénarios sont fréquents en hydrogéologie, il est important de développer des méthodes qui pourraient saisir de manière adéquate l'incertitude dans le sous-sol. À cette fin, cette thèse explore l'adéquation des filtres à particules et des méthodes variationnelles pour l'inférence de l'incertitude des paramètres hydrogéologiques non gaussiens.

Keywords

Hydrogeology • Groundwater • Bayesian statistics • parameter inference • parameter uncertainty • particle filter • data assimilation

---

Hydrogéologie • Eaux souterraines • Statistiques bayésiennes • inférence de paramètres • incertitude des paramètres • filtre à particules • assimilation des données

1 Contents

Acknowledgements

First and foremost, I want to sincerely thank my supervisor Prof. Mario Schirmer, who always supported me in all matters academic and beyond, encouraged me to attend any conferences and workshops I deemed relevant, and allowed me to explore interesting (but not always successful) tangents. I was very fortunate to receive this degree of support – few other PhD students do.

I also owe a great debt to Dr. Carlo Albert, who advised and guided me in all matters mathematical. Without any obligations towards me or this project, he consistently took the time to steer my work through the more turbulent waters of Bayesian statistics. This work has profited tremendously from his advice.

I also want to thank my friends and colleagues from the Water Resources and Drinking Water department: first and foremost Robin Weatherl and Nicole Burri, with whom it was a pleasure to share an office for this journey. I also owe special thanks to our field technician Reto Britt, who diligently maintained our measurement equipment and organized our field sampling campaigns. In addition, I also want to thank Dr. Christian Möck, Marco Dal Molin, and Andrea Betterle and the rest of the department for a great work environment and entertaining coffee breaks. I must also thank Prof. Peter Reichert and Andreas Scheidegger from the SIAM department, who always had an open ear for any theory- or implementation-related questions.

Furthermore, I am grateful to Prof. Matteo Camporese and his work group at the University of Padova, who hosted me for secondment. I also owe thanks to Prof. Philip Renard and Dr. Julien Straubhaar of the University of Neuchâtel, who supported me in the implementation of

their multi-point statistics software. I also want to express my gratitude to Prof. Olaf Cirpka, University of Tübingen, and Prof. Philip Brunner, University of Neuchâtel, who agreed to be part of my PhD committee and took the time to diligently review my thesis. Furthermore, I am grateful to the European Union, who funded this project as part of the Marie Skłodowska-Curie grant agreement No 675120, and all the members of the INSPIRATION ITN.

Finally, I want to thank my friends and family who patiently endured monologues about the more obscure topics of my work. You contained your enthusiasm admirably, but it probably won't get any better in the foreseeable future.

1. Introduction

1.1 About models

"If each city is like a game of chess,
the day when I have learned the rules, I shall finally possess my empire,
if I shall never succeed in knowing all the cities it contains."

- Kublai Khan, in Italo Calvino's *Invisible Cities*

Today, mathematical models permeate almost every aspect of life. Meteorological models predict tomorrow's weather (Pielke, 2013), air traffic models prevent in-flight collisions (Glover & Lygeros, 2004), pharmaceutical models predict the effects of new drugs before clinical test trials (Kumar et al., 2006), and deep neural networks trade on the world's stock exchange markets (Trippi & DeSieno, 1992). Algorithms predict our health, taste in music, and probability to default on a credit. Models are everywhere. Acknowledging that their inner workings can seem (and sometimes be) somewhat arcane or menacing, it may be wise to begin this dissertation with a brief reflection about what models are.

In essence, a model is nothing more than an approximate representation of reality, a mental crutch or a conceptual simplification which allows us to establish causal relationships between concepts. Whether intuitive, conceptual, or mathematical, our thinking is permeated by such models of the mechanisms governing the world around us. In its broadest interpretation, these range from the unconsciously intuitive (*Reading this dissertation barefoot will not increase the number of your toes*) through common knowledge (*Grey skies suggest it might rain*) to advanced scientific deductions (*Seasons change because the planet's rotation axis is not orthogonal to its orbit's plane, varying local surface exposure to the sun*).

As an example of an everyday application of a largely unconscious model, consider the following statement to be true: *The Eiffel Tower is currently located in Paris*. A small amount of time will now have passed since you read the preceding sentence, so you should have no way of answering the question "*Where is the Eiffel Tower now*?" with certainty[1]. The reason you cannot do so is that it demands information you are not given: The statement above only pertains to the location of the Eiffel Tower in the past, about five seconds ago. You may be nonetheless tempted to answer: "*In Paris.*" and obtain a (hopefully) correct prediction, but in doing so you have employed a number of implicit assumptions from the axiomatic *Statements about the past can carry information about the present* up to the object-specific *Massive steel buildings don't suddenly relocate*. In a sense, you were using a mental model of the world, using laws you deem correct, to make a prediction about the real world.

Since these mental constructs are invariably simplifications of reality, the usefulness and reliability of their predictions may vary dramatically, and can often provide useful predictions for entirely the wrong reasons. Early Abrahamic scripture, for example, assumed the wrath of god behind failure to maintain basic sanitation standards around campsites (*Deuteronomy 23, 12-14*). While the predictions made by this scripture are undoubtedly useful, most modern scholars would hesitate to accept this explanation, rather attesting the dire consequences of low sanitation standards to the wrath of E. Coli[2].

[1] Provided you are not currently within eyesight of this landmark. And can trust your eyes. And neglect the possibility that thieves replaced it with a convincing duplicate. And…

[2] Although, lacking a concept of intestinal bacteria and with more pressing matters on their minds, one may forgive the afflicted for their failure to appreciate the difference.

This issue renders the search for knowledge highly challenging, as the flaws in a model may not be immediately obvious. In pursuit of elusive truth and a better understanding of the laws which govern the universe, we must thus constantly revise our conceptual models and construct better hypotheses about the world.

The cosmological model of the universe, for example, is one which has been revised many times. Possibly the first, rather poetic instance of a physical representation of a conceptual model precedes even the old testament and can be found as early as 2000 years BCE in the Epic of Gilgamesh. In this story, Uta-Napishti[3] recounts the construction of his ark to Gilgamesh. According to legend, this ark was a wooden cube of seven floors with nine chambers each, thought to mirror (Maul, 2015) the seven-layered, gemstone-based domes which the Babylonians believed comprised the heavens (Nemet-Nejat, 1998), and the nine realms into which they divided the world[4].

Naturally, the Babylonian model of the universe did not stand the test of time. As new information is made available, these models often required revision. About 500 BCE, the Hellenistic world discovered the spherical shape of the earth (Evans & Jones, 2000). Through the following centuries, several attempts were made to replace geo-centrism with helio-centrism, until Copernicus (Fraser, 2006) finally popularized this view in the 15th century. This model was further adopted and adjusted, eventually leading to the relativistic and quantum-mechanical interpretations we favour today.

[3] Uta-Napishti is the original version of what would later become Noah in the Abrahamic religions, escaping death due to quarrelling gods, and eventually achieving immortality due to what amounts to a contractual loophole. It's a great story.

[4] It seems the Babylonian poet favored symbolism over seaworthiness (Maul, 2015).

The nature of the universe has not changed significantly since the Babylonians first looked at the night sky and sought an explanation that seemed to them both intellectually satisfying and sufficiently intuitive. The process of revising our understanding of the world, then, is as much one of obtaining new insights as it is of learning which question to ask.

1.2 Hydrogeological models

Unfortunately, it is not always possible to generate only a single, unique plausible hypothesis for the observations made. This challenge is of great relevance in hydrogeology, the science of the flow of subterranean water. Groundwater comprises the largest resource of liquid freshwater on the planet and is as such critical to human civilization. Due to its societal and environmental importance, groundwater requires careful management if a stable water supply is to be maintained.

This process is hampered by the difficulty to quantify subterranean resource: it is in the nature of the beast that groundwater is quite literally buried. The subsurface is a highly complex and heterogeneous domain, shaped by millennia of gradual geological processes, complex sedimentology, and progressive physical, chemical, and biological erosion. The result is an environment whose properties can scarcely be derived from the surface, and even less so mapped definitively.

Working in such an environment, then, is an exercise in controlled inadequacy. The management, remediation, and research of groundwater resources demands information which is often not readily available, indirect in nature, or only of limited spatial representativeness. As a consequence, it falls to numerical models to fill the gaps left by field studies and geophysical measurement campaigns with information which is more readily available: data on system *states* such as time-series

of groundwater tables or water chemistry. The task of modelling, then, is to weave plausible narratives within the confines of the observed data. This process is known as inverse modelling.

Considering the exceptional complexity of the system, however, these narratives or hypotheses will rarely be unique: There will usually exist multiple, functionally equivalent explanations for the observed data. Where uncertainty cannot be dispersed, it must be managed, and as such requires a framework for its rigorous management.

1.3 Bayesian statistics

A possible solution to deal with uncertainty may be found in *Bayesian statistics*, which formalizes the process of comparing multiple adversarial hypotheses mathematically. This framework employs the concept of *uncertainty* to quantify the (im)plausibility of any specific hypothesis by combining its *prior* belief (*How plausible does this explanation seem to begin with?*) with its *likelihood* (*Assuming this explanation is correct, how plausible are the observations we made?*) to obtain the posterior (*How plausible does this explanation seem now?*). In many cases, it is also necessary to normalize the product of prior and likelihood with the model evidence (*How plausible is the data [given the model we consider]?*). For discrete or categorical cases, in which the number of possible hypotheses is finite, this is a relatively straightforward and intuitive process (Figure 1-1).

When the number of possible hypotheses becomes infinite, however, the situation is somewhat more challenging. To better understand this case, let us first introduce the idea of a *parameter space*. In principle, a parameter space is no different from the three-dimensional space we live in. However, instead of x, y, and z dimensions, each principal direction

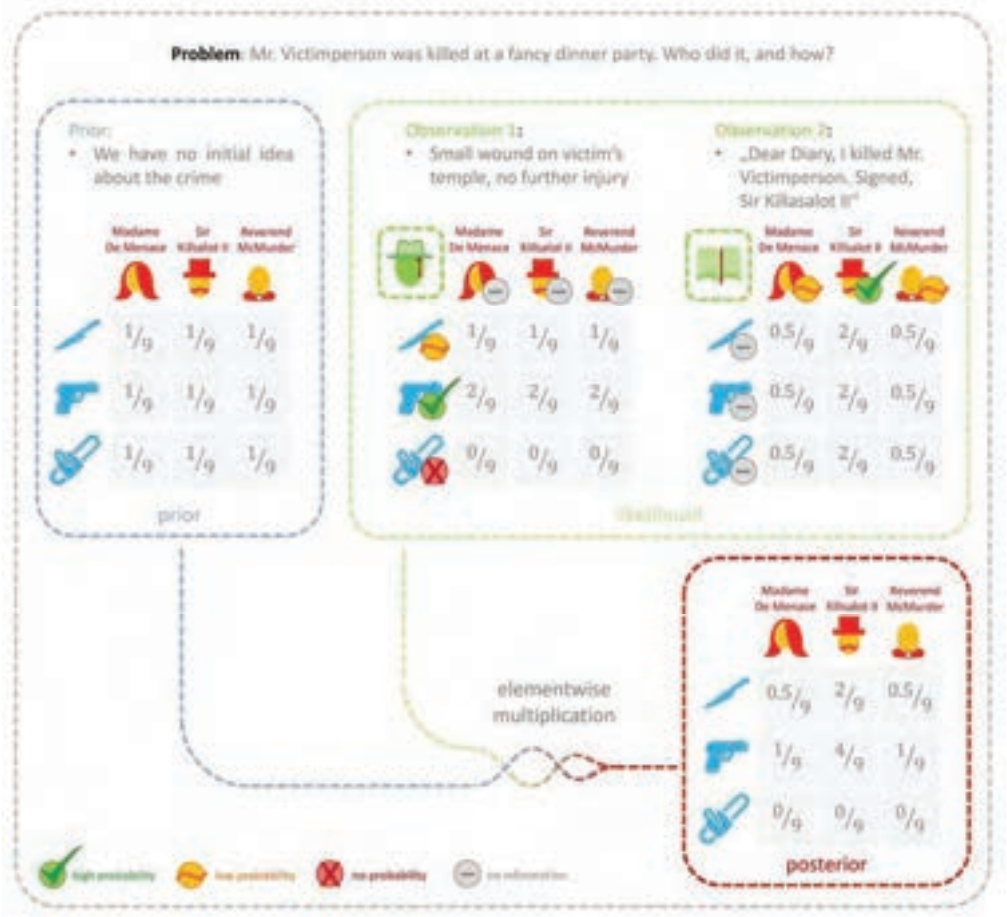


Figure 1-1. Bayesian inference for discrete probabilities as exemplified by a murder at a dinner party. We wish to find the culprit and the murder weapon. One of our variables is the culprit (Madame de Menace, Sir Killsalot II, or Reverend McMurder). The second variable is the murder weapon (a knife, a gun, or a chainsaw). The grey fields represent the probability of a specific combination of suspect and weapon. We start off with an uninformed prior (i.e., all nine options have equal probability). We then make two observations: First, we see that the victim has a small wound on his temple, which suggests that the murder was most likely committed with the gun, possibly with the knife, but definitely not with the chainsaw. This observation does not tell us anything about the culprit. The second observation is a confession by Sir Killsalot II, which suggests that he might be guilty but leaves the possibility that the confession was a forgery by one of the other suspects. This time, the observation does not tell us anything about the weapon. If we combine the prior and the two pieces of likelihood by elementwise multiplication, we obtain posterior probabilities which suggest our most likely explanation is that Sir Killsalot II committed the murder with the gun, but we note that other explanations are also possible.

corresponds to an unknown variable[5], and moving along this direction explores different values or states this variable might take on. A point in this parameter space has coordinates in each of its dimensions and thus represents a hypothesis with a unique combination of variables[6].

[5] For the example in Figure 1-1, we would have a two-dimensional parameters space: the first dimension for the possible suspects, the second dimension corresponds to the possible weapons.

[6] The top-right coordinate in the parameter space of Figure 1-1, for example, has the coordinates 'Reverend McMurder' and 'Knife'.

Even within a finite-dimensional parameter space (one with only a limited number of variables), the realm of the infinite looms as soon as one of its dimensions becomes *continuous*. Examples of continuous variables are height, weight, speed, temperature: values which can vary smoothly along a gradient or have no upper or lower limit. In such cases, it is suddenly no longer possible to write down all hypotheses, and the relatively simple calculation of Figure 1-1 becomes an infinitely large task.

In such cases, considering each individual hypothesis becomes an impossible task. Instead, we must investigate the rules which connect them. Mathematically, this is done through the use of *functions*. In Bayesian statistics with continuous variables, such functions are called *probability density functions*, generally shortened to *pdf*s. As the name implies, these functions assign to each of the infinitely many hypotheses a probability *density*. We talk about densities since an infinite number of hypotheses has to share a finite probability *mass* of 1. Regions of high probability density then correspond to the most plausible regions (Figure 1-2).

These probability density functions have a difficult task: Not only must they reproduce the correct prior probability density for each possible and impossible hypothesis, they must also lend themselves to adaptations in order to reflect the re-allocation of probability density during the learning process. This usually means that the pdf itself has to change, and often even has to change its structure. To reflect arbitrary probability density distributions requires a great degree of flexibility. While it may be possible to find approximate solutions in low-dimensional parameter spaces, this approach quickly becomes computationally untenable in high-dimensional cases with many unknown variables. As such, it is not

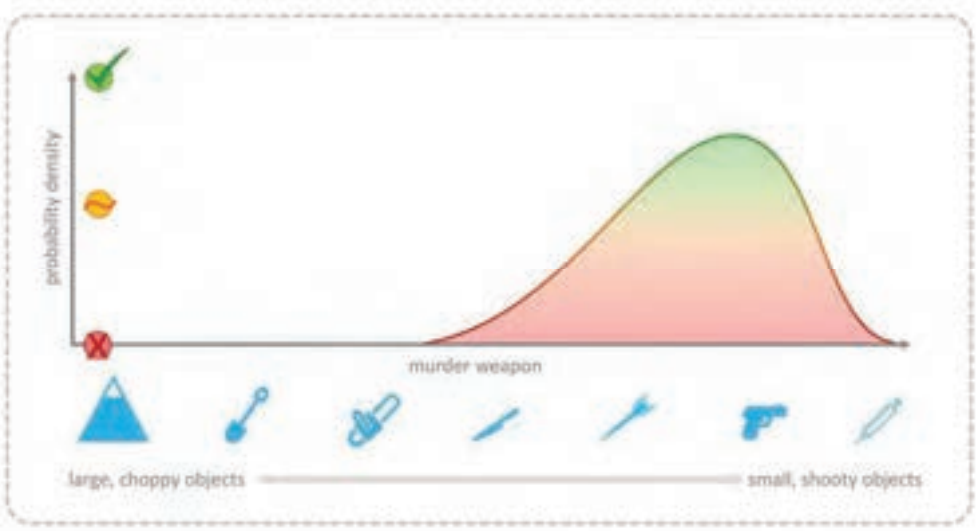


Figure 1-2. Probability density for likelihood of Observation 1, if this parameter would be a continuous variable for possible (and impossible) murder weapons, ranging continuously from large and choppy (a mountain) to small and shooty (a needle).

possible to find a closed-form solution for Bayesian inference in the general case.

There are, however, a few notable exceptions to this rule. Prime among them is the Gaussian pdf (Figure 1-3). In the special case that all pdfs involved are Gaussian, we can solve Bayesian inference analytically: the product of two Gaussian functions will always yield another Gaussian, and transforming a Gaussian linearly also yields another Gaussian. This makes the Gaussian family of functions attractive for Bayesian inference. Unfortunately, in many scenarios the assumptions of Gaussianity cannot be fully justified. In such cases, it thus becomes necessary to consider non-Gaussian features of the pdfs involved.

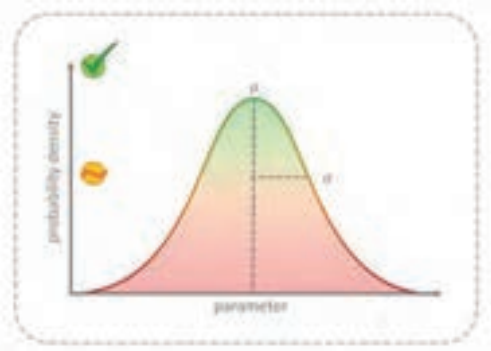


Figure 1-3. Gaussian probability densities are defined by two statistical moments: the mean μ and the standard deviation σ.

1.4 Non-Gaussianity

Gaussians are not a one-size-fits-all distribution and thus have a number of properties which can be problematic in various scenarios. In this introduction, we will mainly focus on the function's support and its unimodality. Other problematic factors include their thin tails (i.e., the low probability assigned to outliers), which can be highly problematic in subjects like economy (Stoyanov et al., 2011), and their simple, global correlation structure (i.e., the value of variable *A* affects the value of variable *B* in a fixed way).

1.4.1 The limits of full support

Among Gaussianity's properties is *full support*[7], which means that these distributions cannot assign zero probability density (i.e., *impossibility*) to any hypothesis within the parameter space. This is problematic for variables with strict physical limits. Every parameter space dimension extends from negative to positive infinity, but certain variable types like concentrations, for example, cannot ever adopt negative values. As such, the chance to obtain a concentration of, say, −5 mg/l should be zero. It would fall to the pdf to assign zero probability density to these values.

This situation can be partially remedied in two ways: by moving to a space where the support is full again (e.g., the logarithmic scale for strictly positive variables), or truncation (Figure 1-4). In the latter approach, one simply defines an upper and/or lower limit for the Gaussian pdf, then re-scales the probability densities within the prescribed limits so that the integral over parameter space still yields a

[7] The support of a function is the subset of its possible input which does not map to zero. For the special case of probability density functions, this means the set of all *possible hypotheses*: those with nonzero probability density.

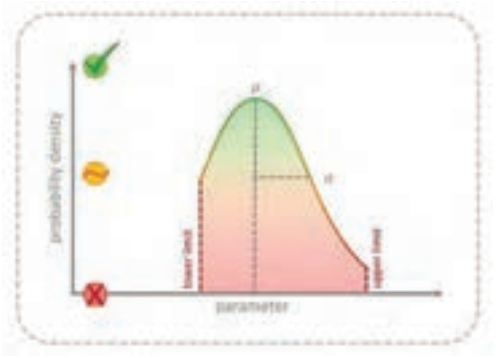


Figure 1-4. A truncated Gaussian pdf.

probability mass of 1. This approach trades mathematical elegance for an often more physically reasonable *convex* [8] support.

Unfortunately, many problems extend beyond even this support. Particularly some of the more complex models require highly elaborate priors which may not lend themselves to a simple mathematical formulation. In hydrogeology, for example, we often discretize the model domain with sometimes up to millions of model cells, each of which can in principle take on parameter values independent from its neighbours (and thus constitutes its own dimension in parameter space). Naturally, we usually need not to assume that each parameter varies independently – if we find sand at one location, chances are high that we will also find sand 5 m in any other direction. Regrettably, this spatial correlation does not generalize very well. The correlation structures between the grid parameters are often locally varying and complex, since they are based on geological or sedimentary formations: ancient riverbeds, glacial moraines, clay lenses, bedrock fractures, and similar features. These features may display sharp discontinuities, whose location may not be necessarily known.

Methods to navigate such complex supports will be discussed in the later chapters. An example of why the support of such priors is not usually

[8] Convex support means that all points on a straight line between any two points within the support set are also supported. Cubes, spheres, pyramids, and similar shapes are all examples of geometric shapes which would constitute a convex support. A donut, or any other shape with holes or a non-convex hull, would not have convex support.

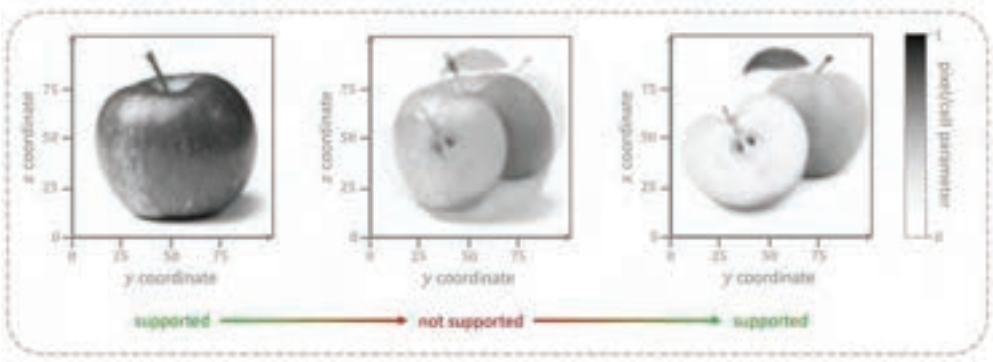


Figure 1-5. Non-convex support for complex priors. In this case, the model domain is parameterized by 10,000 cells (or pixels), each of which can individually take on a value between 0 (white) and 1 (black). Our prior set contains of all pictures showing some form of apple. The centre image, located halfway between the outer images in parameter space, is not in the prior set, and thus not supported.

convex is illustrated in Figure 1-5. In this example, we consider a simple model grid of 100-by-100 cells, where each cell can take on a value between 0 (white) and 1 (black). The resulting parameter space is thus $100 \times 100 = 10{,}000$ -dimensional, and its support convex (each dimension has a lower limit of 0 and an upper limit of 1). To intuitively understand the true scale of this parameter space, interpret the model domain as a screen which can theoretically depict any greyscale image of one's desire in a 100-by-100 resolution: next week's winning lottery numbers, infinitely many images of the reader riding into the sunset on a unicorn, and just plain white noise are all possibilities within this support.

We now want to limit ourselves to a certain prior with highly complex support, say the set of all parameter space dimensions which happen to correspond to an image of an apple (Figure 1-5). In hydrogeology, we would of course instead have geological maps of paleo-channels, moraines, and so forth, but the same principle still applies. If we now consider a position between two different supported points (Figure 1-5, centre), it is unlikely that this point would be supported. Ways around this conundrum are explored in Chapter 2 and Chapter 3.

1.4.2 The limits of unimodality

Gaussian distributions feature only a single *mode* or *optimum*. From a practical perspective, Gaussianity implies that there is a single most plausible hypothesis (located at the mean). All other hypotheses are assigned lower probability densities based on their (Mahalanobis) distance from the mean.

This assumption can be problematic because in many situations there is not just a single clear optimum. Many hydrogeological models, for example, are mathematically underdetermined, which means that there are infinitely many possible parameter combinations resulting in the same model predictions. As a simple example, consider the model $f(a, b) = a + b$ with parameters a and b. Assume that the optimal prediction of this model is $a + b = 4$. It is clear that there are infinitely many solutions which would yield the desired result (e.g., $a = 2$, $b = 2$, or $a = 139.84$, $b = -135.84$). The result would be a so-called *Pareto front* in parameter space. Without an informative prior (and sometimes even then), we cannot distinguish between optimal solutions.

More deceptive – and possibly more dangerous – are cases in which there exist multiple, distinct modes to the pdf, separated by stretches of low probability density. In such cases, the inadequacies of Gaussianity's unimodal assumption may not be immediately obvious, as the inference algorithm may snap towards a single mode. In doing so, however, we risk neglecting functionally equivalent hypotheses which may better describe the true process. These issues are explored in more detail in Chapter 2 and Chapter 4.

1.5 Monte Carlo approximations

While it is important to consider these deviations from Gaussianity from a theoretical perspective, the loss of its elegant mathematical solution is a dear price to pay. Fortunately, there exists methods which permit at least an approximate solution in such cases: the so-called *Monte Carlo* approximation.

The idea behind this technique is relatively simple: If we cannot formulate the pdfs themselves, we can work with *samples* from the pdfs instead. For an intuitive example, let us draw a parallel between a pdf and a dice, in that both generate random samples according to some (in case of the unknown pdf: hidden) rule. Now imagine that we roll this dice repeatedly, writing down each result (a number between 1 and 6) on a piece of paper, and store it in an urn (Figure 1-6). If we repeat this process sufficiently often, we eventually no longer need the dice: we could just as well draw from our collection of samples in the urn and retrieve similar samples[9]. Similarly, we can estimate properties of the dice, such as its mean, its standard deviation, or other statistical moments from the samples it has generated.

While it is often possible to generate samples from complex priors without a well-defined analytical form, the real challenge lies in Bayesian inference. Where we would normally transform the pdfs to reflect a change in knowledge, we must now mimic this process with our collection of samples: Were we to load the die in a manner that it now rolls six twice as often as one, we would have to adjust our samples in the urn as well. We would have to take a third of all samples with 'one'

[9] Somewhat unsurprising, since the samples in the urn were originally drawn from the dice.

Figure 1-6. Schematic illustration of a Monte Carlo approximation. After drawing sufficiently many samples from a pdf or a random process, the collection of samples (ensemble) may act as a surrogate for the true process or pdf.

written on them and change them to 'six', so that there are now twice as many samples with 'six' than with 'one'. In short, we mutate the ensemble of samples in some way so that they become samples of the posterior, or throw away the old samples and draw from the intractable posterior directly. In this dissertation, we will use methods from both classes for non-Gaussian inference.

1.6 Structure of this thesis

This dissertation will explore a number of different Bayesian inference techniques for non-Gaussian scenarios in hydrogeology. The first study

(Chapter 2) discusses the use of particle filters with artificial random dynamics. A small set (called an *ensemble*) of samples (called *particles*), each of which corresponds to a hypothesis, are gradually altered to reflect changes to an underlying, unknown pdf as new information is added (or *assimilated*). The uncertainty information is not contained in a single particle but the ensemble as a whole. The result is an inference algorithm which functions similarly to biological evolution: a small population of model parameterizations attempts to recreate the observations of reality. Well-performing individuals survive and multiply themselves, while ill-performing particles are removed.

A critical part of biological evolution is mutation. In our study, this role is taken by artificial random dynamics, a random component which induce mutations and thus re-introduces diversity into the ensemble (Figure 1-7). However, instead of altering the grid parameters directly (e.g., the hydraulic conductivity of a specific model cell), we instead alter a set of *hyperparameters*. These hyperparameters are input to a deterministic (i.e., non-random) field generator, which in turn creates the grid parameters which ultimately affect the model predictions.

This has the advantage that we can guarantee conformance to prescribed geological features[10]. The drawback is that the relationship between the hyperparameters we update and the states (like water tables or concentrations) can become more nonlinear if the field generator is nonlinear. This can be problematic for algorithms like the Ensemble

[10] Borrowing from before: if grid parameters can be thought of as a TV screen, our field generator could be thought of as a puppet theatre. Assuming we want scenes with a policeman and a crocodile, both the TV screen and the puppet theatre could theoretically create them. However, if we would randomly shake the system (assuming shaking a TV would randomly re-arrange its pixels), only the puppet theatre would consistently create scenes with a crocodile and a policeman. If we would subsequently like to convert these random scenes to a TV screen (grid parameterization), we could just use a camera (field generator) to film the puppet theatre (hyperparameterization).

Kalman Filter, which rely on a linear causal relationship between parameters and states. However, for particle filters and biological evolution alike – both of which rely on random mutation instead of gradient information – this is not a major obstacle.

Favourable results show that this algorithm can successfully optimize a hydrogeological model but cannot retain diversity in the ensemble for long – likely a consequence of highly-tapered posteriors and high-dimensional parameter spaces. As a consequence, the uncertainty information in the ensemble is eventually lost, as the diversity in each 'generation' collapses to clones of a single individual. Furthermore, while this algorithm shares strengths of biological evolution, it also shares some of its weaknesses: mutations which have proven beneficial in the past are eventually 'forgotten' without constant reinforcement. Just as our biological ancestors' ability to survive in an aquatic environment has been lost to time, the model may forget adaptations exclusively beneficial for the summer months during winter. Strengths of the algorithm are favourable optimization performance in a synthetic setting, the ability to navigate complex prior support in parameter space through the use of hyperparameterization, and limited computational effort which permits the use for online (*quasi-real-time*) optimization.

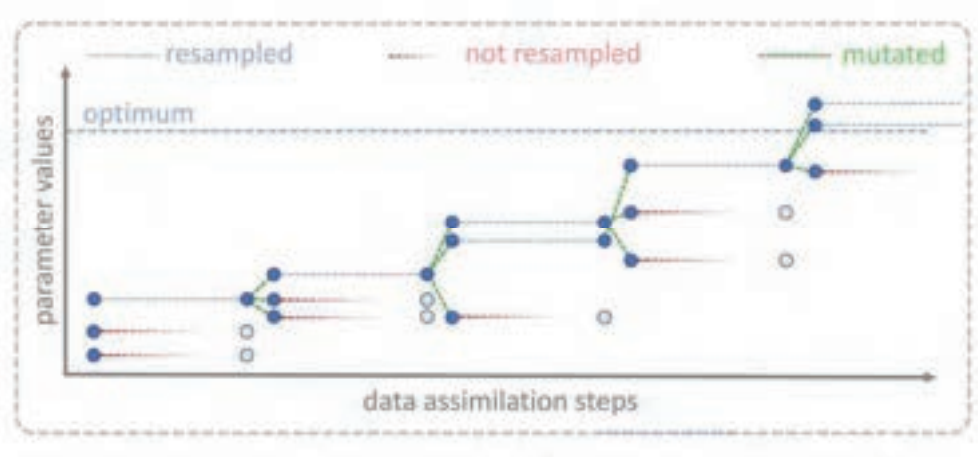


Figure 1-7. Schematic illustration of a particle filter with an artificial random dynamic. It functions similar to biological evolution: the mutations are random, but the selection process is not. The result is a 'blind' optimizer.

To address some of the limitations of the first study, the second study (Chapter 3) explores a variation of this evolutionary algorithm. While still based on a selection-mutation scheme intrinsic to most particle filters, the mutations no longer occur indiscriminately. Instead, they are comprised of Monte Carlo Markov Chain (MCMC) jumps: mutations are always accepted if they constitute an improvement but are only sometimes accepted if they are not (Figure 1-8). From a practical perspective, this approach has the advantage that it rigorously tests if particles constitute an improvement, as opposed to evaluating only their most recent performance, but the price for this is steadily increasing computational effort as the timeseries of data grows. Furthermore, since soil parameters are usually invariable on the time scales of interest, this guarantees that states (such as water tables) and parameters (such as hydraulic conductivity) are always physically consistent with each other. We combine the algorithm with a complex geological field generator based on multi-point statistics – an algorithm which can generate variations of a training image – and apply it to a real field site in Northern Italy. Our results show favourable optimization performance in

comparison to the Ensemble Kalman Filter but the loss of diversity in the ensemble – and thus of parameter uncertainty – still remains a significant limitation.

The strengths of this approach are the physical consistency between states and parameters and the ability to optimize multi-point statistics realizations. Unfortunately, since this method is also based on random mutations, the loss of uncertainty remains an unaddressed challenge.

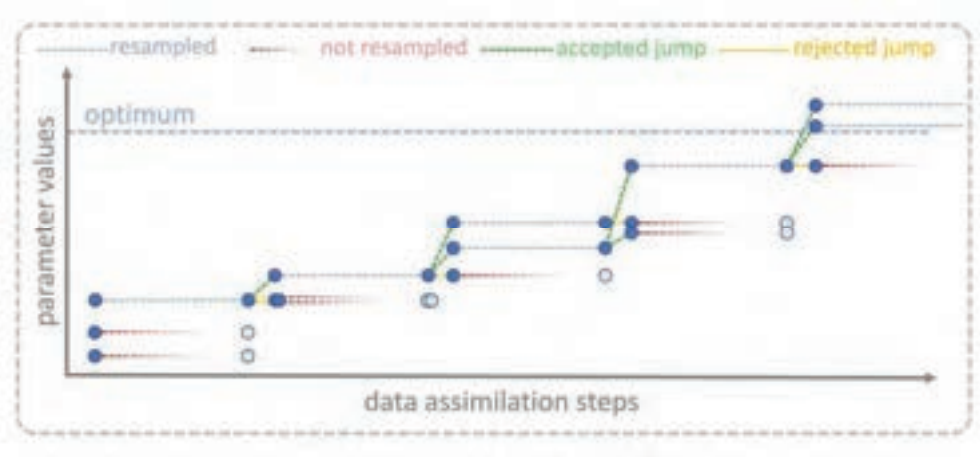


Figure 1-8. Schematic illustration of a particle filter with MCMC rejuvenation jumps (Iterated Batch Importance Sampling). As opposed to artificial random dynamics, mutations are no longer accepted indiscriminately, although they are still random.

In this dissertation's final study (Chapter 4), we explore an alternative, novel algorithm which may circumvent the issue of ensemble collapse: Stein Variational Gradient Descent (*SVGD*: Liu & Wang, 2016). Instead of discarding bad particles and refreshing (or *rejuvenating*) the ensemble through random mutation, all ensemble members compare the quality of their adaptations, and then evolve themselves into promising directions (in our implementation: towards better-adapted particles, and away from worse-adapted ones).

In a sense, this seems similar to the highly robust Gaussian-based Ensemble Kalman Filter (*EnKF*: Evensen, 1994, 2003), which transforms each particle based on ensemble-based information. Crucially, however,

SVGD does not rely on the assumption of Gaussianity, and where the EnKF adds information incrementally, SVGD follows an iterative scheme[11]. Together, the ensemble thus moves towards the optimum while retaining diversity at all times (Figure 1-9).

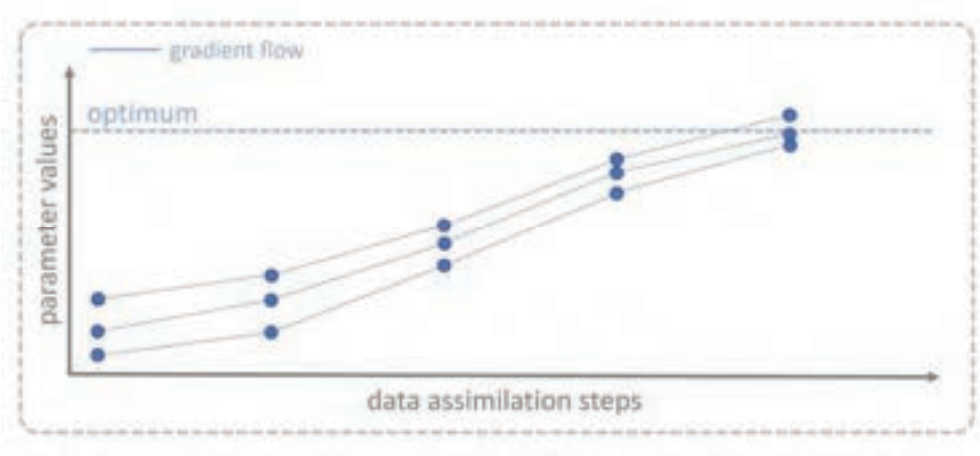


Figure 1-9. Schematic illustration of Stein Variational Gradient Descent. This approach no longer uses a mutation/selection scheme, but instead iteratively and smoothly transforms each particle into a sample of the posterior.

Due to its iterative nature, this algorithm does not particularly care about the 'sharpness' of the tapered posterior, and can even split the ensemble to follow multiple alternative optimization pathways at once. Furthermore, it can mirror the EnKF's ability to optimize on a subspace of parameter space. If the ensemble consists of N particles and the Jacobian is estimated from the ensemble, the particles move in an at most $N-1$-dimensional subspace. Note that this property is a devil's bargain: it implicitly scales the complexity of the inference problem with the computational resources available. This can be useful because it means the algorithm performs well even when only few particles are available (a virtually omnipresent scenario in hydrogeology), but carries

[11] Think *navigation system* rather than *treasure map*: The former homes you in on the target, whereas the latter requires you to follow its steps exactly.

significant risks, since it limits the space of solutions and implicitly 'hides' the true complexity of the inference problem.

The SVGD algorithm is derived and tested for a simple synthetic and a complex real hydrogeological test case in Chapter 4. We found promising optimization performance in both cases, and believe that this algorithm and its relatives are a highly promising direction for future research. A limitation of this algorithm is its restriction to convex support, which means that many of the complex geological priors (see Chapter 1.4.1) cannot be optimized without moving to a space with convex support first. Despite this, it is still a significant improvement over Gaussianity, as the resulting posterior doesn't have to be unimodal Gaussian, and the model can be highly non-linear.

In Chapter 5, the conclusions of this dissertation will be presented, and promising directions for future research suggested. The references of this dissertation are listed in Chapter 6, followed by the supporting information for the individual chapters.

2 Data Assimilation and Online Parameter Optimization in Groundwater Modelling using Nested Particle Filters[12]

2.1 Abstract

Over the past decades, advances in data collection and machine learning have paved the way for the development of autonomous simulation frameworks. Among these, many are not only capable of assimilating real-time data to correct their predictive shortcomings, but also of improving their future performance through self-optimization. In hydrogeology, such techniques harbour great potential for informing sustainable management practices. Simulating the intricacies of groundwater flow requires an adequate representation of unknown, often highly heterogeneous geology. Unfortunately, it is difficult to reconcile the structural complexity demanded by realistic geology with the simplifying assumptions introduced in many calibration methods. The particle filter framework would provide the necessary versatility to retain such complex information, but suffers from the curse of dimensionality, a fundamental limitation discouraging its use in systems with many unknowns. Due to the prevalence of such systems in hydrogeology, the particle filter has received little attention in groundwater modelling so far. In this study, we explore the combined use of dimension-reducing techniques and artificial parameter dynamics to enable a particle filter framework for a groundwater model. Exploiting

[12] This chapter has been published in Water Resources Research: Ramgraber, M., Albert, C., & Schirmer, M. (2019). Data Assimilation and Online Parameter Optimization in Groundwater Modeling Using Nested Particle Filters. *Water Resources Research*. https://doi.org/10.1029/2018WR024408.

freedom in the design of the dimension-reduction approach, we ensure consistency with a predefined geological pattern. The performance of the resulting optimizer is demonstrated in a synthetic test case for three such geological configurations and compared to two Ensemble Kalman Filter setups. Favourable results even for deliberately mis-specified settings make us hopeful that nested particle filters may constitute a useful tool for geologically consistent real-time parameter optimization.

2.2 Introduction

Parameter estimation is an essential part of any simulation in which the demand for parameter information outweighs its availability. In few environmental disciplines is this discrepancy as pronounced as in hydrogeology: any attempt to faithfully capture the intricacies of groundwater flow demands a realistic representation of often highly heterogeneous geology (Rubin & Hubbard, 2006). Unfortunately, the extent, nature or even existence of these features can scarcely be derived from the surface (e.g., de Marsily et al., 2005). Direct measurements to ascertain their properties offer little relief, constituting a time- and resource-intensive endeavour with no guarantee to adequately delineate the flow-relevant structure (Schöniger et al., 2012).

The task of parameter estimation is the inference of the properties from information of dependent states. In hydrogeology, transient hydraulic heads often take this role. This topic has been explored in a large body of literature over the years (e.g., Hill et al., 2000; McLaughlin & Townley, 1996; Yeh, 1986). Among these techniques, *batch-calibration* (or *history-matching*) approaches – which rely on bulk-processing a pre-existing set of observations – have been state of the art for decades and still remain highly popular today. Many of these techniques, like the parameter

estimation and uncertainty analysis tool PEST (John Doherty, 2015; John Doherty et al., 2010), have become wide-spread industry standard.

More recently, however, there has been a growing interest in the development of *real-time* (or *online*) parameter estimation techniques (e.g., Hendricks Franssen & Kinzelbach, 2008). Sparked by increasing availability of real-time data from wireless sensor networks (e.g., Cardell-Oliver et al., 2005) and satellite-based remote-sensing (e.g., Houser et al., 1998), such algorithms assimilate a data stream of state measurements to gradually improve parameters during active model operation. The advantage over batch-calibration approaches is evident: there is no need to wait until a sufficiently large body of data is collected. Instead, such algorithms may autonomously process data as they become available, theoretically providing a best-guess estimate at all times.

2.2.1 Limitations of the state of the art

In pursuit of data assimilation and real-time calibration, the *Ensemble Kalman Filter* (*ENKF:* Evensen, 1994, 2003) has established itself as one of the most popular approaches in environmental science. The reasons for this algorithm's success are manifold: ease of implementation (Iglesias et al., 2013), high computational efficiency (Hu et al., 2013; Zovi et al., 2017), and relative robustness to violations of its fundamental assumptions of *Gaussianity* and *linearity* (Iglesias et al., 2013; Katzfuss et al., 2016) equally contributed to its prevalence in the data assimilation community.

Nonetheless, the EnKF is not without shortcomings: its core assumptions are essentially never met in hydrogeological practice, and in such cases its high efficiency comes at the price of excluding many possible solutions. When mean and variance do not provide sufficient statistics

for the probability density function (*pdf*) under investigation, or the system propagation is non-linear, the assumption of Gaussianity may be violated and the EnKF will yield only approximate solutions (e.g., Amezcua & Van Leeuwen, 2014; Schillings & Stuart, 2017), the usefulness of which has to be evaluated on a case-by-case basis.

Unfortunately, the pursuit of realistic geology during parameter estimation is a case in which these approximations often prove insufficient. Parameter uncertainty due to ignorance of geological and sedimentary features can take on several shapes: often, not only the hydraulic properties of these features are unknown, but also their spatial extent and arrangement. From a mathematical perspective, the uncertainty originating from the unknown arrangement of the geological features is characterized by distinct multimodality of the parameters' pdf and thus poorly reflected by the EnKF's assumption of unimodal Gaussianity. It has been shown that if a latent geological structure of the prior is not sufficiently informed by the observed states, the EnKF updates tend to not (or only vestigially) preserve its characteristics (e.g., Zovi et al., 2017).

In a bid to address this issue, it has been proposed to employ *Gaussian anamorphosis* (*GA*, sometimes also called *normal score transformation*), an approach converting non-Gaussian marginal distributions to unimodal Gaussian ones for the duration of the assimilation step (e.g., Schöniger et al., 2012; Zhou et al., 2011). While GA has been reported to alleviate this structural degeneration to a certain degree (Zovi et al., 2017), it may come at the price of increased non-linearity of the observation operator (Amezcua & Van Leeuwen, 2014) or the relationship between transformed variables (Zhou et al., 2011). When applied only to state space, Schöniger et al. (2012) observed a more linear relationship

between transformed states and untransformed parameters. However, they remark that univariate transformations may only transform the marginals but cannot alter the multivariate dependence structures between the variables. For some types of data (e.g., concentration data; see Schöniger et al., 2012) this dependence structure can be far from Gaussian and in such cases univariate transformations will not yield multivariate Gaussian values.

An interesting alternative approach was proposed by Hu et al. (2013). Rather than calibrating the model parameters directly, their EnKF implementation filters white noise fields instead. These noise fields are subsequently used as random seeds in the generation of geostatistical parameter fields from multi-point statistics (*MPS*: e.g., Caers & Zhang, 2005). Since MPS generates fields consistent with a pre-defined geology – the training image –, their EnKF implementation retains consistency with the desired structure throughout model calibration. Unfortunately, such indirect approaches introduce (further) non-linearity. This might affect the filter's performance, since the convergence of EnKF-based parameter estimation depends on the strength of the linear correlation between state observations and filtered parameters (Jafarpour & Tarrahi, 2011).

2.2.2 Beyond Gaussianity

In an effort to overcome the assumption of Gaussianity and its limitations, the particle filter is a natural alternative to the Ensemble Kalman filter. Based on a direct Monte Carlo representation of the underlying pdf, this filter makes no assumptions about the shape of the pdf or the nature of the system dynamics (e.g., Doucet & Johansen, 2009; Doucet & Tadić, 2003) and updates its ensemble through adjustments of

the particles' retrieval weights. Unfortunately, this flexibility comes at the price of the so-called *curse of dimensionality*: the number of Monte Carlo samples (*particles*) required to adequately represent the pdf increases exponentially with the number of unknown variables (e.g., Bengtsson et al., 2008; Farchi & Bocquet, 2018; van Leeuwen, 2010). Hydrological numerical models may have millions of cells, and each cell may have different hydraulic parameters, so that naive applications of the particle filter have been deemed computationally infeasible (e.g., Aanonsen et al., 2009; Ruiz et al., 2013; Schöniger et al., 2012). Over the past decades, particle filters have been assigned a niche role in hydrology, largely limited in their application to conceptual or lumped models with only few unknown parameters (e.g., Moradkhani et al., 2005). Recently, their scope of application has widened to include distributed models such as drought forecasting frameworks (Yan et al., 2017, 2018).

However, much progress has been made in the development of highly efficient filter techniques (Morzfeld et al., 2017). Particularly for state estimation, the number of required particles can be drastically reduced: Through manipulation of the proposal pdf, van Leeuwen (2010) reports a successful application of a filter with only 20 particles to a 1000-dimensional problem – an otherwise thoroughly hopeless task – and surmises that "the curse of dimensionality may have a cure". Recently, van Leeuwen et al. (2019) give an overview of particle filter variants for high-dimensional geophysical applications, mostly focused on state estimation, and Farchi & Bocquet (2018) provide an overview over local particle filters, suggested by Snyder et al. (2008) to overcome obstacles of high-dimensional particle filters. For parameter optimization, some authors (e.g., Abbaszadeh et al., 2018; Zhu et al., 2018) transferred

elements from genetic algorithms to particle filters in order to alleviate the curse of dimensionality.

In general, the raw number of unknown variables may be a poor measure of the system's true complexity. Especially in hydrogeology, the required number of model parameters may be significantly smaller than the number of model cells. Such considerations already form the basis for dimension-reduction in hydrogeological parameter estimation (e.g., J Doherty & Hunt, 2010), and would naturally extend to particle filtering. For more complex priors, the size and shape of the geological features can be uncertain, too, so that geometrical parameters may need to be included.

In the following sections, we will provide a brief conceptual overview of the particle filter and its limitations and present a formal derivation and algorithmic implementation of the nested particle filter. With the framework established, we will discuss hyperparameterization as a tool to exploit latent structural simplicity in the numerical grid. To conclude, we will illustrate the performance of such an algorithm for different geological conceptualizations using synthetic examples and compare it to results obtained from an EnKF.

2.3 Theory and Methods

In this section we will provide a general introduction to particle filters and their limitations. Subsequently, we will present a formal derivation of the nested particle filter algorithm and a blueprint for its implementation. We conclude this section with a brief discussion of hyperparameterization and artificial parameter dynamics. First, however, we will introduce the nomenclature used in this study.

2.3.1 Nomenclature

In many filtering applications a distinction is made between *parameters* and *states*. Parameters, in this work denoted $\boldsymbol{\theta}$, are usually static model variables, such as hydraulic conductivities or porosities, properties which generally do not depend on other variables. States, represented by $\boldsymbol{x}$, are often time-varying system variables depending on parameters and model forcings: hydraulic heads, temperatures, or concentrations all are common examples. Observations are treated as a third variable type, $\boldsymbol{y}$, and are generally measurements of states. For implementation-related purposes, all model variables of a type are combined into a vector and interpreted as coordinates of a point in high-dimensional space (*parameter space* or *state space*, respectively). *Particles* occupy one such point, and thus represent a full set of their respective variables required for a model. Individual particles, or variables related to them, are assigned a superscript index in brackets, e.g. $\boldsymbol{x}^{(index)}$. We use a point instead of an explicit index, e.g. $\boldsymbol{x}^{(\cdot)}$, if we refer to all indices of a given type. Specific time points are denoted by subscripts $\boldsymbol{x}_{time}$, and ranges between a start and end point are represented by $\boldsymbol{x}_{start:end}$. A '$\sim$' should be read as 'sampled from'.

2.3.2 The particle filter

Like many other parameter optimization approaches, the particle filter is based on a probabilistic framework. It serves as a tool to represent uncertainty about (and inability to perfectly replicate) what we assume is an unknown, but fundamentally deterministic natural system. In a probabilistic framework, variables are not assigned a single value ('*the porosity is 0.24*') but can theoretically take on all mathematically possible values ('*the porosity is somewhere between* $-\infty$ *and* $+\infty$'). Since this

formulation carries no specific information, knowledge or belief about the plausibility of different variables is specified exclusively by a *probability density function* (*pdf*) defined over all numeric values ('*plausible porosity values lie between 0 and 1 with these probability densities*'). In such a framework, any change in knowledge of the system must be reflected in a change of the corresponding pdf. This process is formally described by Bayes' theorem:

$$p(A|B) = \frac{p(B|A)\,p(A)}{p(B)} \qquad 2\text{-}1$$

Bayes' theorem updates the prior pdf $p(A)$, representing the belief about a given parameter predating an observation B, to its more informed posterior pdf $p(A|B)$. This operation is achieved by introducing the likelihood $p(B|A)$ of the observation B for a given value A. In order to ensure that the posterior probability density function $p(A|B)$ integrates to unity the numerator has to be normalized by the marginal likelihood $p(B)$.

In the general case it is practically impossible to pursue this Bayesian inference analytically. As a consequence, one may either confine the analysis to a special case like the Kalman filter, which is restricted to Gaussian priors and likelihoods so that an analytic solution is available, or to forfeit an analytic solution altogether and approximate the pdfs involved through an ensemble of weighted, deterministic Monte Carlo samples (e.g., Arulampalam et al., 2002).

The key idea behind the latter approach is that an unattainable analytic target distribution $p(A)$ may be approximated by a set of N deterministic independent and identically distributed (i.i.d.) samples thereof

$$\hat{p}(A) = \frac{1}{N}\sum_{i=1}^{N} \delta_{A^{(i)}}(A) \xrightarrow{N\to\infty} p(A) \qquad 2\text{-}2$$

where $\delta_{A^{(i)}}(A)$ is the Dirac delta distribution centred about $A^{(i)}$, with $(i) = 1, \ldots, N$ being the particle's index. This superposition of Dirac delta distributions illustrates the surrogate properties of the Monte Carlo set: Instead of sampling $p(A)$, one could (at least in the limit of $N \to \infty$) equivalently draw with probability $\frac{1}{N}$ from a sufficiently large, pre-existing set of realizations thereof.

Particle filtering, then, is the technique of recursively updating this set of realizations to retain its surrogate properties along otherwise intractable Bayesian inference operations. Provided one could ensure that the ensemble of Monte Carlo samples remains representative of $p(A|B)$, it is theoretically possible to proceed indefinitely without the need to ever recover an analytic expression. These recursive updates are achieved with a procedure based on importance sampling; by adjusting the particles' retrieval weights w_i (initially uniform $\frac{1}{N}$) a set of particles drawn from one pdf could approximate a different distribution altogether. It should be intuitive that this re-weighting is useful in the context of Bayesian inference: the posterior $\hat{p}(A|B)$ may be approximated by the particles of the prior, $\hat{p}(A)$. Employing the nomenclature of importance sampling one can interpret $\hat{p}(A)$ as the importance distribution, $\hat{p}(A|B)$ as the nominal distribution, and the normalized likelihood as the importance weight. This yields:

$$\hat{p}(A) = \sum_{i=1}^{N} w_i \delta_{A^{(i)}}(A) \qquad 2\text{-}3$$

with weights w_i and $\sum_{i=1}^{N} w_i = 1$.

In theory, this operation could be repeated indefinitely, continuously adjusting the particle weights to reflect the latest posterior. In practice, however, this approximation will become increasingly inefficient and without an infinite ensemble size eventually only one particle will retain any significant weight. To counteract this *particle degeneracy*, it is common practice to resample the particles Figure 2-1.This duplicates highly-weighted particles but leads to a loss of variation in the ensemble (*sample impoverishment*).

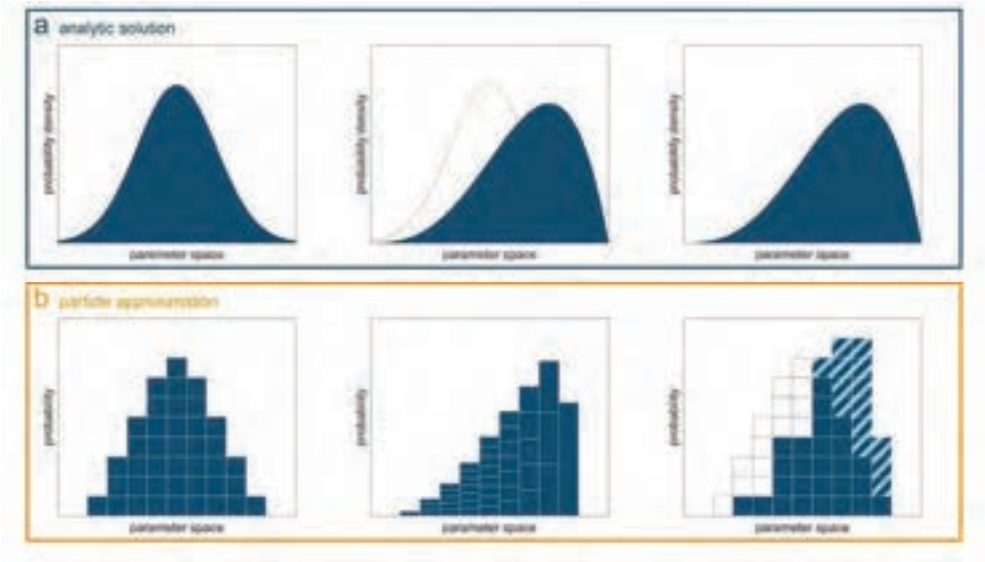


Figure 2-1. Conceptual scheme of Bayesian inference with a particle filter. The lower row (b) depicts a histogram representing the particle approximation of the analytic solution depicted in the upper row (a). First, the prior is approximated by equally weighted i.i.d. samples, visualized as blocks (left); during Bayesian updating, the particles are re-weighted (illustrated by adjusting the block heights) so that the ensemble approximates the posterior pdf (middle); finally, new samples may be drawn from the weighted particles to yield a new, equally-weighted ensemble, resetting the cycle (right).

In transient stochastic systems, this loss of diversity is counter-acted by the forecast's random component, mapping duplicated state particles to slightly different positions. For the physically often static parameters, however, no such dynamic exists. Common approaches to re-introduce diversity to the parameters (*rejuvenation*) include MCMC steps (e.g., Chopin et al., 2013), jittering, or a combination of the two. In our study we will follow the second path by explicitly defining artificial parameter

dynamics, which we will outline in Section 2.3.4. First, however, we introduce the specific particle filter used in this study.

2.3.3 The nested particle filter

The key idea behind a nested particle filter is to separate the filtering of states and parameters (e.g., Chopin et al., 2013; Dan Crisan & Miguez, 2013). This is achieved in a 'hierarchical' manner, whereby a single outer particle filter performs parameter inference, while several inner particle filters – one assigned to each parameter particle – conduct the state inference. This arrangement constitutes the nested structure and provides valuable information to both variable types: the state particle filters inherit their parent's parameter values and may treat them as fixed for the forecast, and the parameter particle filter can evaluate its particles' otherwise intractable likelihoods.

It may be worthwhile to note how the nested particle filter relates to the classic particle filter approach for joint state-parameter estimation in hydrology. By reducing the ensemble size of the inner filter to a single particle, one would retrieve the more commonly used particle filter with an augmented state vector (e.g., Montzka et al., 2011; Moradkhani et al., 2005). Larger inner ensemble sizes trade computational efficiency for better parameter likelihood estimates and can furthermore enable the use of more sophisticated particle filter algorithms (see van Leeuwen et al., 2019 for examples) which may not trivially extend across parameter space.

As explained in Section 2.3.2, the introduction of new diversity is crucial to the reversal of ensemble collapse. Since numerical models are never perfect replications of reality, knowledge about states should degrade during the forecast between successive inference steps. In particle filters,

this loss of information (increase of entropy, pushing the pdf towards uniformity) is generally represented by diffusion through an additive random error component, which – from a practical perspective – rejuvenates the particles. For the physically static parameters, however, no such natural source of diversity exists. To avoid non-reversible ensemble collapse, we employ artificial parameter dynamics in this study. We note that this source of noise is introduced out of computational convenience rather than reflecting a physical process, and account for it explicitly in the filter's derivation in the following section. The consequences of and opportunities created by artificial parameter dynamics are explored in more detail in Section 2.3.4. In this section, we will first present the formal background of the nested particle filter with time-varying parameters, followed by its algorithmic implementation.

2.3.3.1 Formal justification

The core objective of the algorithm presented in this study is the estimation of model parameters $\boldsymbol{\theta}$ given a predefined geological characterization. Due to the introduction of noise through artificial parameter dynamics, these parameters become a time-varying quantity described by their trajectory through time $\boldsymbol{\theta}_{0:m} := (\boldsymbol{\theta}_0, \dots, \boldsymbol{\theta}_m)$. Further introducing a sequence of state observations $\boldsymbol{y}_{1:t} := (\boldsymbol{y}_1, \dots, \boldsymbol{y}_t)$ and a likelihood function $p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}_{1:m})$ allows for basic Bayesian inference:

$$p(\boldsymbol{\theta}_{0:m}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{\theta}_{0:m})p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}_{1:m})}{p(\boldsymbol{y}_{1:t})} \qquad 2\text{-}4$$

where the discrete time step indices $c = 0, \dots, m$ and $s = 1, \dots, t$ illustrate the possibility for parameter and state dynamics, respectively, to operate on different time scales. In this instance we define parameter dynamics slower than its state counterpart, with each cycle c being comprised of L state time increments. The likelihood of the observation trajectory

conditional on the parameter trajectory $p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}_{1:m})$ is a property that can be difficult to infer directly. Instead, one may expand the numerator by introducing the evolution of the predicted model state vector $\boldsymbol{x}_{0:t} := (\boldsymbol{x}_0, \dots, \boldsymbol{x}_t)$ to the likelihood and integrating it out immediately. We further recognize that the denominator is the integral of the numerator over the parameter trajectory, whose purpose is the normalization of the numerator, allowing us to formulate the expression to proportionality ($\propto$). This yields:

$$p(\boldsymbol{\theta}_{0:m}|\boldsymbol{y}_{1:t}) \propto \int p(\boldsymbol{\theta}_{0:m})p(\boldsymbol{y}_{1:t}, \boldsymbol{x}_{0:t}|\boldsymbol{\theta}_{1:m})\, d\boldsymbol{x}_{0:t} \qquad 2\text{-}5$$

Reformulating the likelihood in terms of the observation trajectory conditional on the trajectory of the predicted model states yields

$$p(\boldsymbol{\theta}_{0:m}|\boldsymbol{y}_{1:t}) \propto \int p(\boldsymbol{\theta}_{0:m})p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{1:t})p(\boldsymbol{x}_{0:t}|\boldsymbol{\theta}_{1:m})\, d\boldsymbol{x}_{0:t} \qquad 2\text{-}6$$

Now, we may introduce a sequential solution by defining cycles $c = 1, \dots, m$, each composed of sub-steps $u = 1, \dots, L$. The state time increments s are replaced by cycle-dependent subscripts $z = (c-1)L + u$. We further introduce the state forecast operator $f(\boldsymbol{x}_z|\boldsymbol{x}_{z-1}, \boldsymbol{\theta}_c)$, the likelihood function $g(\boldsymbol{y}_z|\boldsymbol{x}_z)$, and the artificial parameter dynamics $k(\boldsymbol{\theta}_c|\boldsymbol{\theta}_{c-1})$:

$$p(\theta_{0:m}|y_{1:t}) \propto \int p(\boldsymbol{\theta}_0)p(\boldsymbol{x}_0)\prod_{c=1}^{m}\left\{k(\boldsymbol{\theta}_c|\boldsymbol{\theta}_{c-1})\prod_{u=1}^{L}[f(\boldsymbol{x}_z|\boldsymbol{x}_{z-1}, \boldsymbol{\theta}_c)g(\boldsymbol{y}_z|\boldsymbol{x}_z)]\right\} d\boldsymbol{x}_{0:z} \qquad 2\text{-}7$$

2.3.3.2 Algorithmic implementation

We implement Equation 2-7 as a nested particle filter. As previously elaborated, this approximates the analytically intractable parameter pdf by a set of N_θ particles drawn from the prior, which are then recursively re-weighted, re-sampled and mutated. Equivalently, the state pdf is

approximated by N_θ ensembles of N_x state particles each. Superscripts $(n_\theta) = 1, \dots, N_\theta$ and $(n_x) = 1, \dots, N_x$ refer to specific particles of the parameter and state ensembles. Particles with multiple superscripts, e.g. $\boldsymbol{x}^{(n_\theta, n_x)}$, should be interpreted like this: 'state particle (n_x) of the inner filter belonging to parameter particle (n_θ)'. To aid intuition, we will discuss the implementation subdivided into its two main constituents: the outer particle filter for the parameters, and the inner particle filters for the states.

Inner particle filters

Since each inner particle filter is assigned to a 'parent' parameter particle, there are a total of N_θ such filters in parallel. In the following, we will consider only one of these filters as a blueprint for all others and hence assume its parent particle $\boldsymbol{\theta}_c^{(n_\theta)}$ as given. We start by drawing N_x i.i.d., equally-weighted initial state particles from a suitable prior. This prior does not necessarily have to have an analytic form, but could be obtained from, say, an interpolation of randomly perturbed observations, or steady-state simulations with a probabilistic error.

Then, the state particles are propagated individually via the forecast operator $f\left(\boldsymbol{x}_z^{(n_\theta, n_x)} | \boldsymbol{x}_{z-1}^{(n_\theta, n_x)}, \boldsymbol{\theta}_c^{(n_\theta)}\right)$ to the next time step at which observations are available. In practice, the forecast operator is constructed of two parts: First, the deterministic numerical groundwater model $\boldsymbol{M}\left(\boldsymbol{x}_{z-1}^{(n_\theta, n_x)}, \boldsymbol{\theta}_c^{(n_\theta)}\right)$ which uses $\boldsymbol{x}_{z-1}^{(n_\theta, n_x)}$ as the initial conditions and $\boldsymbol{\theta}_c^{(n_\theta)}$ as the parameters. Uncertain forcings or boundary conditions could also be considered at this point but are assumed known in this synthetic study for the sake of simplicity. The second part is an error term $\varepsilon_{model} \sim \mathcal{N}(0, \sigma_{model}^2)$ which, in our case, consists of a homogeneously applied Gaussian noise with mean 0 and variance σ_{model}^2:

$$\boldsymbol{x}_z^{(n_\theta,n_x)} = \boldsymbol{M}\left(\boldsymbol{x}_{z-1}^{(n_\theta,n_x)}, \boldsymbol{\theta}_c^{(n_\theta)}\right) + \mathcal{J}\varepsilon_{model} \quad 2\text{-}8$$

where $\mathcal{J} = (1, \dots, 1)^T$ is a vector of ones with the same dimensions as the vector $\boldsymbol{x}$. The choice of this error term is not trivial to determine and thus often left to the modeler's discretion. The choice of error we made above bears two advantages over other commonly used error types:

(i) A homogeneous error may only alter large-scale absolute deviation in the model's state budget, but has little influence on relative state distributions, thereby reducing interference of the model error on the characteristic flow responses of different parameter particles to a minimum: A sufficiently large spatially-correlated error (for example a Gaussian random field) could create artificial gradients in the head field by allowing the filter to repeatedly resample state particles with perturbations which add or remove water in certain regions of the model domain, whereas a sufficiently large spatially-uncorrelated error could cause numerical instability.

(ii) Applying the error homogeneously, on the other hand, restricts the state perturbation onto a diagonal line of slope 1 in state space irrespective of its dimensionality, effectively rendering the error one-dimensional at the cost of limiting the algorithm's ability to 'correct' state predictions.

Once all state particles are propagated and a new observation vector $\boldsymbol{y}_z$ is obtained, the particles can be weighted. First, determine the likelihood $g\left(\boldsymbol{y}_z|\boldsymbol{x}_z^{(n_\theta,n_x)}\right)$ of the observations conditional on the predictions for each updated state particles. Assuming independent normal observation errors $\varepsilon_{obs} \sim \mathcal{N}(\mu = 0, \sigma_{obs}^2)$, the full likelihood function can be treated as a composite likelihood (e.g., Varin et al., 2011)v calculated from the

product of N_{obs} likelihoods – one for each individual observation. Each such component is evaluated according to:

$$l_z^{(n_\theta,n_x,o)} = \frac{1}{\sqrt{2\pi\sigma_{obs}^2}} exp\left(-\frac{\left(x_z^{(n_\theta,n_x,o)} - y_z^{(o)}\right)^2}{2\sigma_{obs}^2}\right) \quad 2\text{-}9$$

where superscript $(o) = 1, \dots, N_{obs}$ denotes an index of a specific observation in the observation vector y_z or the index of the corresponding prediction in the state vector $\boldsymbol{x}_z^{(n_\theta,n_x)}$. The composite likelihood $\ell_z^{(n_\theta,n_x)}$ for each state particle is calculated as the product over all independent likelihood components:

$$\ell_z^{(n_\theta,n_x)} = \prod_{o=1}^{N_{obs}} l_z^{(n_\theta,n_x,o)} \quad 2\text{-}10$$

This composite likelihood can be used to determine the un-normalized weight of the corresponding state particle $\boldsymbol{x}_z^{(n_\theta,n_x)}$. In this study, we use the prior as the proposal distribution (Chopin et al., 2013) but note that other choices are possible (e.g., van Leeuwen, 2010). Assuming the prior weights are equal, we can normalize the likelihoods to retrieve normalized weights:

$$w_z^{(n_\theta,n_x)} = \frac{\ell_z^{(n_\theta,n_x)}}{\sum_{n_x=1}^{N_x} \ell_z^{(n_\theta,n_x)}} \quad 2\text{-}11$$

At the end of each data assimilation step, the state particles are resampled. We select N_x new state particles from the multinomial distribution weighted according to $w_z^{(n_\theta,\cdot)}$ (Gordon et al., 1993) (Figure 2-1). After each state particle $\boldsymbol{x}_z^{(n_\theta,n_x)}$ is independently assigned an ancestral index $a = 1, \dots, N_x$, it copies the states of its respective ancestor $\boldsymbol{x}_z^{(n_\theta,a)}$, and the state particle weights are reset to uniformity. The filter

then proceeds to the next time step increment $z+1$ and repeats from Equation 2-9.

Outer particle filter

Similarly to the state particle filters, we initiate the parameter particle filter by drawing samples from a suitable prior. Like the inner particle filters, the process starts with mutating the particles. As previously established, this is achieved through the artificial parameter dynamics $k(\boldsymbol{\theta}_c|\boldsymbol{\theta}_{c-1})$. The precise nature of these dynamics will be explored in detail in Section 2.3.4, but assume for now that they yield a slightly mutated particle $\boldsymbol{\theta}_c^{(n_\theta)}$ for each progenitor $\boldsymbol{\theta}_{c-1}^{(n_\theta)}$:

$$\boldsymbol{\theta}_c^{(n_\theta)} \sim k\left(\boldsymbol{\theta}_c^{(n_\theta)}|\boldsymbol{\theta}_{c-1}^{(n_\theta)}\right) \qquad 2\text{-}12$$

The next step is the evaluation of the likelihoods, which are extracted from composite likelihoods of the inner particle filters $\ell_z^{(n_\theta,\cdot)}$ before the state particles are resampled. Since each cycle is composed of L sub-steps $u=1,\dots,L$, we form the product of the composite likelihoods over the current cycle, then form the Monte Carlo integral over the inner particle filter to retrieve the cycle's marginal likelihood $\mathcal{L}_c^{(n_\theta)}$ for each parameter particle:

$$\mathcal{L}_c^{(n_\theta)} = \frac{1}{N_x}\sum_{n_x=1}^{N_x}\prod_{u=1}^{L}\ell_z^{(n_\theta,n_x)} \qquad 2\text{-}13$$

Assuming prior equal weights this marginal likelihood may serve as an un-normalized weight for the parameter particles, and yields normalized weights according to:

$$w_c^{(n_\theta)} = \frac{\mathcal{L}_c^{(n_\theta)}}{\sum_{n_\theta=1}^{N_\theta}\mathcal{L}_c^{(n_\theta)}} \qquad 2\text{-}14$$

Finally, the weighted parameter particles are resampled equivalently to the procedure of the inner particle filters. Each resampled particle inherits not only its ancestor's parameter values, but also its ancestor's state particle filter. After resampling, the filter proceeds to the next cycle $c + 1$ and repeats from Equation 2-13. The pseudocode for the algorithm is provided in Figure I-1 (supporting information).

2.3.4 Artificial parameter dynamics

Introducing artificial parameter dynamics carries the explicit assumption that the model parameters evolve (randomly) in time. This approach is not a new idea (e.g., Doucet & Tadić, 2003; Li et al., 2014) and has been used in a variety of applications (e.g., D. Crisan & Miguez, 2018; Kitagawa, 1998). Some EnKF variations, such as the one suggested by Pathiraja et al. (2018), also employ explicit random parameter dynamics for active covariance inflation. Other EnKF inflation methods such as damping of the analysis (Hendricks Franssen & Kinzelbach, 2008; Keller et al., 2018) or linear scaling (Anderson & Anderson, 1999) achieve a similar effect deterministically. Even if no covariance inflation is used, practically all EnKF applications with joint parameter inference nonetheless render the parameters time-varying through the analysis step, although this is rarely stated. A notable exception to this is the Restart EnKF (Gu & Oliver, 2007), designed specifically to address this issue by re-starting the simulation with a time-constant parameter set following the analysis step.

Rendering otherwise static parameters time-varying is a potentially dangerous assumption: In general, these artificial dynamics will have no physical equivalent and may thus risk leading to inconsistencies between states and parameters. Time-varying parameters can, however,

be justified in sufficiently dissipative settings which tend to 'forget' their history and thus prevent the accumulation of error (groundwater flow, following a diffusion equation, is generally dissipative).

We note that alternative approaches to address ensemble collapse in particle filters exist, such as the Metropolis-Hastings Markov chain Monte Carlo jumps employed by the SMC² algorithm (Chopin et al., 2013). This technique does not require the assumption of time-varying parameters and is invariant with respect to the target distribution but comes at the cost of steadily increasing computational effort, rendering it ill-suited for online application.

Beyond these considerations, however, the introduction of artificial dynamics is attractive in several ways. Primarily, it permits the parameter particle ensemble to rejuvenate itself by introducing a form of covariance inflation. Cycles of resampling followed by slight mutations allow the ensemble to gradually explore regions of parameter space not sampled by the initial particles. Unfortunately, an efficient exploration of parameter space based on random mutation is still precluded by the curse of dimensionality. Hyperparameterization offers a way to alleviate this issue and provides additional interesting opportunities we will explore in the following.

Hyperparameterization

Recall that it is often possible to reduce the effective dimensionality of systems, for example by principle component analysis (*PCA*: Wold et al., 1987). In the case of discretized subsurface models, dimension reduction is often achieved through the use of pilot points (e.g., John Doherty et al., 2010; RamaRao et al., 1995), more conventional zonation into geological units (or facies) of shared properties (e.g., Jesus Carrera & Neuman, 1986; Hendricks Franssen et al., 2009; Yeh, 1986), or interpolation techniques

such as kriging, inverse distance weighting or splines (e.g., Robinson & Metternicht, 2006; Yeh, 1986). These reduced parameter sets are able to generate the fully-dimensional parameter field required by the model via a pre-defined set of rules. They can thus be interpreted as *hyperparameters* – parameters describing other parameters. The set of rules by which the hyperparameters relate to the full parameter field equivalent will subsequently be called a *field generator*. An illustration of this process is provided in Figure 2-2.

Adopting a hyperparameterization carries several distinct advantages. First, it allows the user to reduce the dimensionality of the model's parameterization to the degree of complexity demanded by the geological features rather than the numerical grid. The schema in Figure 2-2 illustrates that the number of hyperparameters is substantially lower than the number of parameters (six opposed to 460) and the parameterization becomes independent of grid resolution. More importantly, the field generator guarantees conformance with prescribed geological structures through construction by restricting the exploration of parameter values to a different, possibly simpler space. Conversely, all parameter fields which cannot be created by the field generator are no longer possible outcomes, which may be welcome in pursuit of geological realism. We note that many more sophisticated object-based field generators such as HyVR (Bennett et al., 2019) or FLUVSIM (Deutsch & Tran, 2002) have been developed and could be adapted to interface with optimizers such as the one presented in this study.

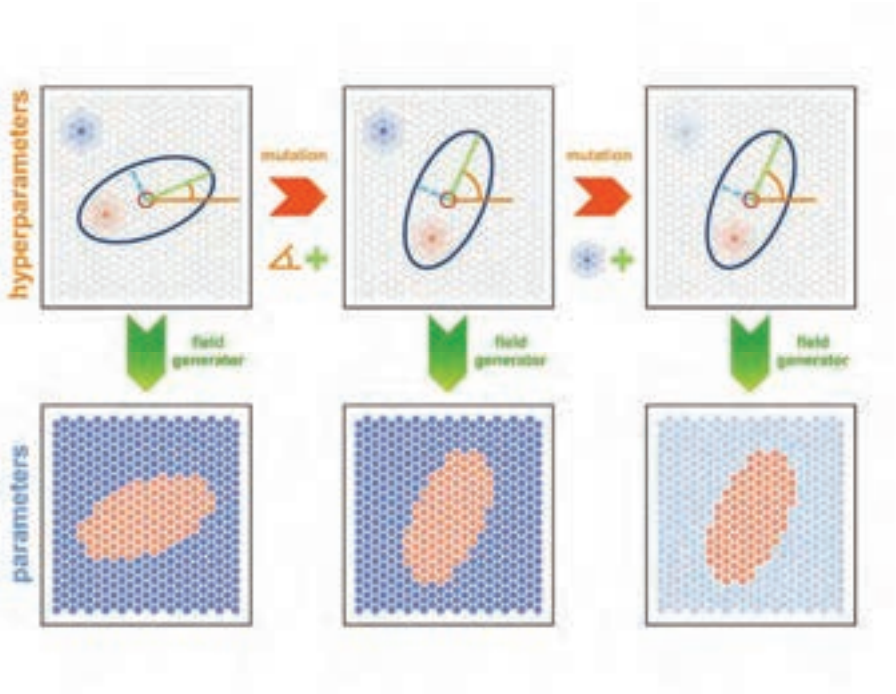


Figure 2-2. Conceptual scheme of hyperparameterization: a limited number of hyperparameters (primary axis length, secondary axis length, position, rotation, two conductivities) can generate a full parameter field via a field generator (green arrows). Randomly mutating the hyperparameters (orange arrows) can efficiently alter aspects of the corresponding parameter field while maintaining the prescribed geological pattern – in this case, a lens embedded in a background sediment.

Random dynamics in hyperspace

The reduced dimensionality of the auxiliary hyper(parameter)space permits a simpler exploration of the parameter space, for example via Gibbs sampling (e.g., Kruschke, 2015). To implement the artificial parameter dynamics, we employ custom *mutation operators* designed to efficiently and randomly explore the hyperspace. These operators are largely based on Gibbs updates, but may include more intricate actions designed to capitalize on pre-defined expectations of system response. Beyond the Gibbs updates, some of the more complex operators like 'swapping node conductivities' (a reflection in hyperspace) find equivalents in mutation operators of genetic algorithms (*interchanging*, *swap* or *Twors* mutation; see e.g., Abdoun et al., 2012; Sivanandam & Deepa, 2008). Others like 'removing lenses' or 'adding nodes' (a removal or addition of hyperspace dimensions) are only rarely encountered in evolutionary algorithms, since most problem statements assume a fixed

dimensionality of the parameter space (Lee & Antonsson, 2000). Formally incorporating hyperparameterization into the framework derived in Section 2.3.3 could be achieved in two ways:

(i) to entirely replace the parameters with hyperparameters, and interpret the field generator as a deterministic part of the numerical model (a hyperspace-based perspective); or

(ii) to incorporate them as a 'latent logic' underlying the artificial random dynamics in parameter space (a parameter space-based perspective). Both views are different interpretations of the same process.

2.3.5 EnKF setups

To provide a comparison to more prevalent algorithms in hydrogeological literature, we also consider two different EnKF setups: a classical augmented state-vector EnKF, and a augmented state-vector EnKF with Gaussian anamorphosis (GA). Both EnKF setups are initialized with parameter fields created by the field-generators but subsequently operate in joint state-parameter space (for the standard EnKF setup) or the transformed joint state-parameter space (for the GA-EnKF setup). This approach is a widely used in subsurface hydrology (e.g., Tang et al., 2015; Zhou et al., 2011; Zovi et al., 2017), often with initial parameter fields generated by MPS. In our study, both setups are implemented with covariance localization following the approach of Hamill et al. (2001)with a length scale d_{loc} of 120 m:

$$\boldsymbol{Z} = \begin{cases} 1 - \frac{1}{4}\left(\frac{d_{ij}}{\lambda}\right)^5 + \frac{1}{2}\left(\frac{d_{ij}}{\lambda}\right)^4 + \frac{5}{8}\left(\frac{d_{ij}}{\lambda}\right)^3 - \frac{5}{3}\left(\frac{d_{ij}}{\lambda}\right)^2 & d_{ij} \le \lambda \\ \frac{1}{12}\left(\frac{d_{ij}}{\lambda}\right)^5 - \frac{1}{2}\left(\frac{d_{ij}}{\lambda}\right)^4 + \frac{5}{8}\left(\frac{d_{ij}}{\lambda}\right)^3 + \frac{5}{3}\left(\frac{d_{ij}}{\lambda}\right)^2 - 5\left(\frac{d_{ij}}{\lambda}\right) + 4 - \frac{2}{3}\left(\frac{d_{ij}}{\lambda}\right)^{-1} & \lambda < d_{ij} \le 2\lambda \\ 0 & d_{ij} > 2\lambda \end{cases} \quad \text{2-15}$$

where $\boldsymbol{Z}$ is the localization matrix, $\lambda = d_{loc}\sqrt{10/3}$, and d_{ij} is the Euclidian distance between cells i and j. Furthermore, both setups employ a Kalman gain damping factor of $\alpha = 0.5$. In combination, we calculate the Kalman gain K as follows:

$$\boldsymbol{K} = \alpha(\boldsymbol{C} \circ \boldsymbol{Z})\boldsymbol{H}^T(\boldsymbol{H}(\boldsymbol{C} \circ \boldsymbol{Z})\boldsymbol{H}^T + \boldsymbol{R})^{-1} \qquad 2\text{-}16$$

where $\boldsymbol{C}$ is the augmented state vector covariance matrix, $\circ$ denotes elementwise multiplication, $\boldsymbol{H}$ is the operator extracting the observation locations, and $\boldsymbol{R}$ is the (diagonal) observation error covariance matrix. For the GA-EnKF, $\boldsymbol{C}$ and $\boldsymbol{R}$ are replaced by their respective transformed equivalents. The GA was implemented following the procedure prescribed in Schöniger et al. (2012), using the anamorphosis function extrapolation rule of Keller et al. (2018) and the observation error covariance transformation approach of Geppert (2015) and transforming both state and parameter spaces, back-transforming after each analysis step.

We further consider a hybrid setup, replacing the nested particle filter's inner filter with EnKFs for the states only, with $d_{loc} = 120\ m$, $\alpha = 0.01$, and no GA. This setup is designed to capitalize on the comparatively linear dynamics and Gaussian uncertainties in state space, while retaining the particle filter's flexibility for the more demanding parameter inference. The state particle likelihoods $\ell_z^{(n_\theta, n_x)}$ required for the outer filters 2-13) are extracted from the inner EnKFs just before the analysis step.

2.4 Synthetic case study

2.4.1 Model setup

We test the algorithm in a synthetic, horizontal 2-D case study allowing for perfect knowledge of the reference 'reality'. The model domain is tessellated with hexagonal grids to maximize structural isotropy while minimizing cell count. The studied aquifer is unconfined, and its synthetic geology features a high-conductive paleo-channel embedded within low-conductive background material (Figure 2-3). Observations of hydraulic head are taken at 16 wells distributed regularly in a 4x4 grid by extracting the corresponding nodal values and adding uncorrelated Gaussian noise with zero mean and a standard deviation of $\sigma_{obs} = 0.02\,m$. The geology was assumed known at only three of these wells. Flow in the system is driven by periodic recharge draining to the western fixed-head (Dirichlet) boundary. This causes the hydraulic heads to approach dynamic steady state after a sufficiently large number of cycles, implying they are transient but time periodic. The initial state particles were generated by individually perturbing the initial state observations with the observation error and inter- and extrapolating to all other cells. Further relevant model and filter properties are listed in Table S1 (supporting information). The algorithm was implemented in Python, interfacing with MODFLOW-USG (USG: unstructured grid, Panday et al., 2013) through the Python package FloPy (Bakker et al., 2016). Parameters to be estimated were hydraulic conductivities, and the observed and predicted states were hydraulic heads.

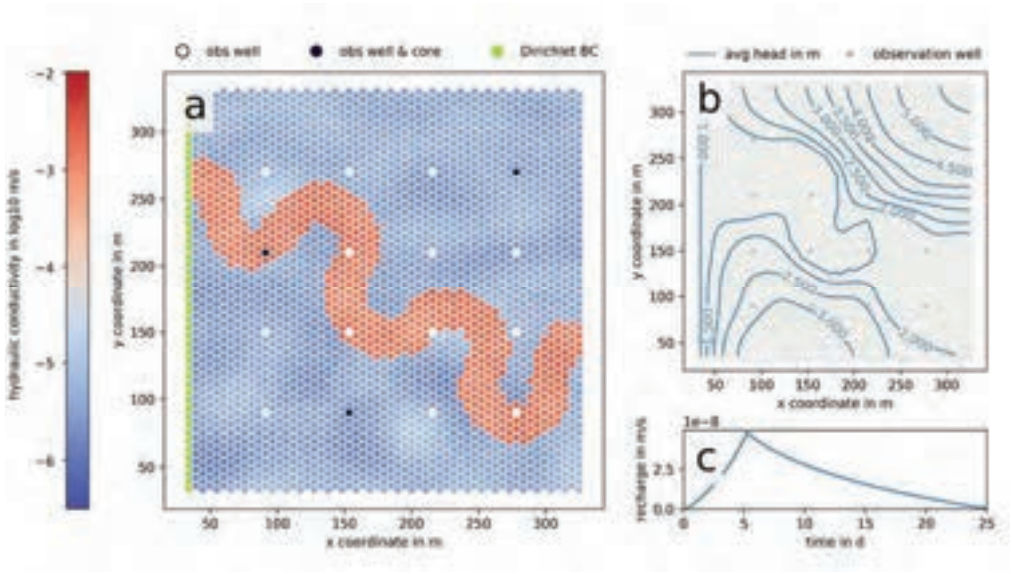


Figure 2-3. Schematic overview of the synthetic reference. The geology consists of a high-conductive paleo-channel in low-conductive background material (a). Observations are taken at 16 locations, but the geology is assumed known at only three wells. The northern, eastern, and southern boundaries are no-flow, while the western boundary with a fixed head of 1 m serves as the system's only sink. The system's mean water table (b) is established by periodic recharge boundary condition (c).

2.4.2 Field generators and mutation operators

We test the nested particle filter algorithm for three hyperparameterizations corresponding to three different geological conceptual models against the same synthetic reference. The motivation for investigating these scenarios is that geological characterizations of field sites are essentially always imperfect. The hyperparameterizations considered are based on pilot points, lenses embedded in background sediment, and a meandering paleo-channel embedded in background sediment. The latter two scenarios feature Gaussian heterogeneity independently within the background sediments and features (channels/lenses), which was deliberately mischaracterized: the correlation length of the synthetic test case is three times larger than its representation in the model used in data assimilation (Figure 2-4).

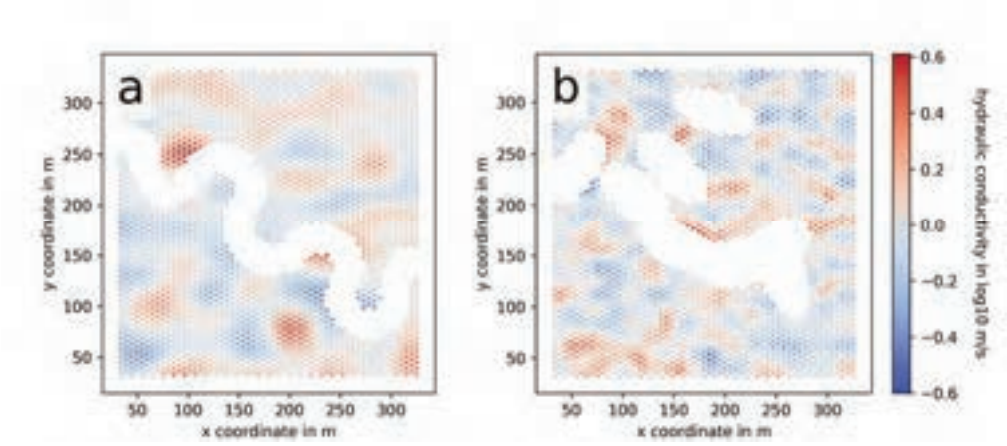


Figure 2-4. Mischaracterization of the synthetic reference's facies heterogeneity. The isotropic correlation length of the Gaussian heterogeneity in the synthetic reference's geology (a) is three times as large as assumed in the geological characterization (b).

Following the concepts introduced in section 2.3.4, each scenario corresponds to a unique hyperparameterization with its own field generator and mutation operator. When called, the mutation operators carry out none (5% chance), one (55% chance) or two (40% chance) of a selection of actions which are listed in Table S2 (node-based), Table S3 (lens-based), and Table S4 (meander-based) in the supporting material. The no-mutation chance is added to ensure that the ensemble can remain within an optimum, once located.

2.4.2.1 Node-based field generator

The first field generator is based on a classical interpolation approach: a selection of nodes is randomly placed in the model domain, each of which is assigned a hydraulic conductivity. Creation of the full parameter field follows from inter- and extrapolation according to the nodes' and cells' spatial positions using inverse distance weighting with a high power factor (Shepard, 1968). This yields parameter fields very similar to what would be obtained through Voronoi tessellation (e.g., Aurenhammer et al., 2013). The hyperparameters are the number of nodes, and the position and hydraulic conductivity of each node. Points of known geology are represented by fixed nodes with immutable,

perfectly known hydraulic conductivity. On average, this scenario features 100 hyperparameters.

2.4.2.2 Lens-based field generator

This field generator creates geological patterns based on distributing elliptical lenses of defined geometric properties over the model domain. Creation of the full parameter field is achieved by assigning each cell one of two sediment facies, depending on its placement with respect to the lenses. Parameters are then assigned based on two separate facies-specific conductivity maps for each particle, each of which is defined over the whole model domain with a specified mean, standard deviation and variogram. The hyperparameters are the conductivity mean for both facies maps, the number of lenses, and the position, size, rotation, and aspect of each lens. Points of known geology are limits enforcing one of the two facies types, whose internal heterogeneity is – as described above – mischaracterized. On average, this scenario features 42 hyperparameters.

2.4.2.3 Meander-based mutation kernel

The third field generator is conceptually similar to the lens-based field generator, but it generates meander-like features. First, the field generator generates a meander from the hyperparameters. The creation of the full parameter field is based on assigning each grid cell a sediment facies depending on whether it is located within or outside the meander. This scenario also employs two facies maps, with the same fundamental mischaracterization of internal heterogeneity. The creation of the meander from the hyperparameters is a multi-step process illustrated in Figure I-2 (supporting information). Hyperparameters are the mean hydraulic conductivity for both facies maps and several parameters specifying the meander: the position and orientation of the start- and

end-points, the meander phase shift, the meander width and the channel width. Points of known geology are represented as guaranteed adherence to one of the two facies types. Due to the complex nature of this field generator, an iterative procedure is required to ensure conformance with points of known geology. After mutation, the meander's phase shift is randomly adjusted until a conforming facies distribution is found. If no such move is possible, the proposed mutation is rejected. This scenario features 9 hyperparameters.

2.4.3 Computational setup

The algorithm was tested on two desktop computers using a 64-bit Windows 7 OS, with Intel® Core™ i7-2600 CPU @ 3.4GHz and Intel® Core™ i7-3770 CPU @ 3.4GHz processors and 8 GB of RAM. The simulation of the full synthetic calibration period of 750 days for the nested particle filter scenarios and the hybrid scenario was achieved using 200 parameter particles with 5 state particles each. For the EnKF scenarios, the ensemble was initialized with 1000 joint state-parameter realizations.

Computation times for the nested particle filter varied depending on the hyperparameterization used: The node- and lens-based scenarios required about 18 core hours for a full run, well below the available time in a field application. The meander-based hyperparameterization runs required significantly more time due to the iterative nature of its mutation operator, demanding about 49 core hours of computation time. The simulation times reported above were obtained for a sequential implementation of the algorithm. A parallelized setup was tested but dismissed as ineffective after increasing the computation time by a factor of 4 due to overhead. Each scenario was repeated with ten different

random seeds (*RS*) to test the reproducibility of the results obtained. A full run for the EnKF setups required about 18 core hours each. This time requirement was constant for all hyperparameterizations, since the field generators were only called once during the initialization of the algorithm.

2.5 Results and Discussion

We first report the parameter fields obtained at the end of the data assimilation period. Subsequently, we investigate the head and conductivity discrepancies between the ensemble results and the synthetic reference. Finally, we investigate the deterministic prediction performance of the obtained parameter fields and conclude with a discussion of the results.

2.5.1 Optimized parameter fields

Figure 2-5 depicts a selection of parameter fields at different points in time as obtained by the nested particle filters. The first column (Figure 2-5a, d and g) illustrates selected samples from the initial parameter fields, created by the different field generators with randomized hyperparameter input. The second column (Figure 2-5b, e, and h) shows the expectation of the final cycle's ensemble for selected random seeds. The third column (c, f, and i) illustrates the average parameter field over all ten random seeds obtained at the end of the calibration period. A comparison with Figure 2 reveals that even the structurally mischaracterized field generators (*node-based* and *lens-based*) seem to evolve their hyperparameters in a way allowing for the creation of a structure functionally similar to the reference meander.

Results for the hybrid nested particle filters with inner EnKFs match the findings of these scenarios (Figures S3 to S5, supporting information). This can be explained by collapse of state uncertainty in the first quarter of the simulation time (Figures S6 to S8, supporting information), after which there are no more functional differences between the two particle filter setups.

Note that the mean parameter fields (Figure 2-5; Figures S3 to S5) are more 'crisp' than one would expect from an ensemble-based optimizer. This results from a weight-based ensemble collapse of the outer filter during every calibration cycle, right before new diversity is created by the artificial parameter dynamics. Contributing factors are the parameter cycle length, the number of observation points, the observation error standard deviation, the sensitivity of the predictions to changes in the parameter values, and the magnitude of parameter changes proposed by the artificial parameter dynamics. While this study's model and setup resulted in repeated collapses, other systems and sites may be less prone to degeneration and could preserve more of the probabilistic information. For interested readers, a quantitative analysis of this collapse considering the setup of the nested particle filter is provided in *Appendix 1: Investigation of the ensemble collapse*. It is worth noting, however, that the optimization of the parameter fields proceeds despite the loss of most probabilistic information. This becomes evident in the posterior parameter fields' tendency to express hydraulic conductivity distributions functionally similar to the synthetic reference field (Figure 2-5).

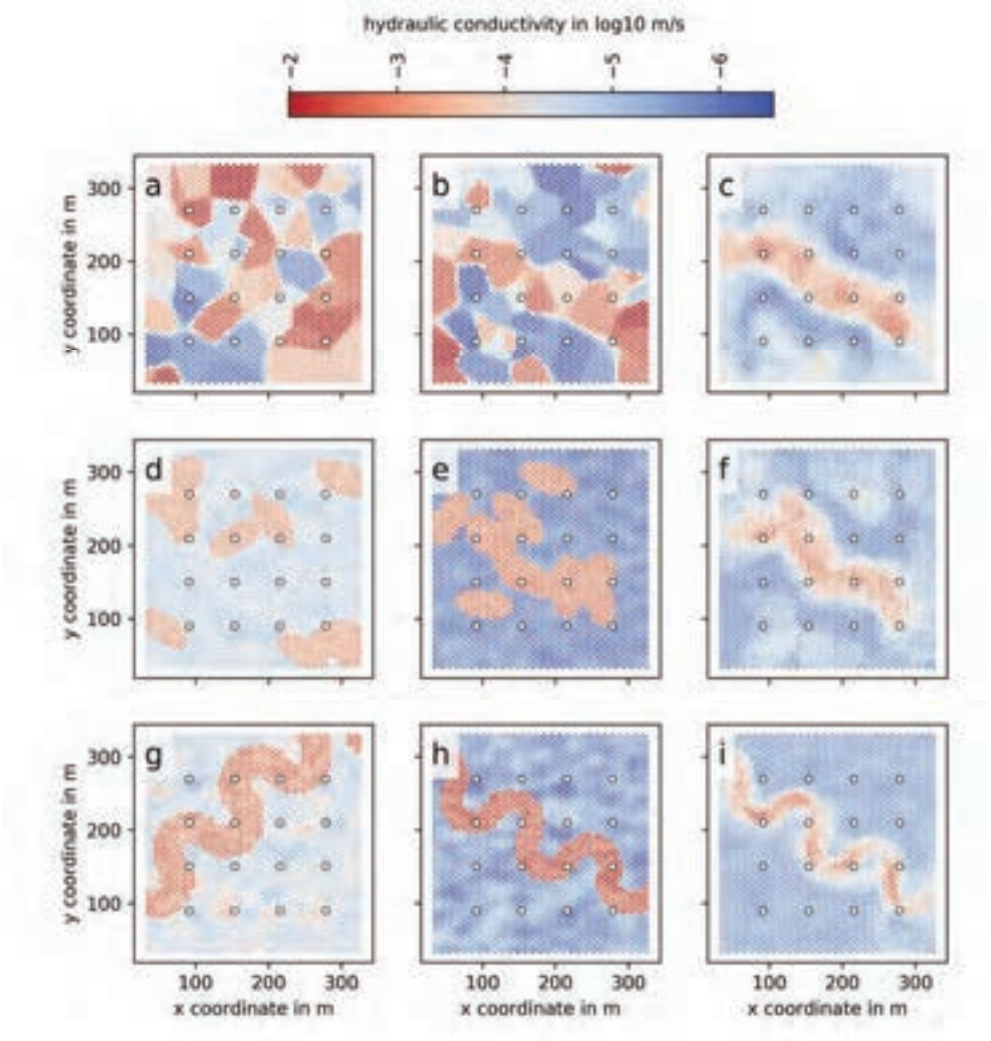


Figure 2-5. Selected parameter fields for the three field generators: node-based (a-c), lens-based (d-f) and meander-based (g-i); the first column (a, d, g) illustrates selected initial parameter particles; the second column (b, e, h) shows expected final parameter fields for selected random seeds; the third column (c, f, i) depicts mean final parameter fields through all ten random seeds.

Figure 2-6 summarizes the results for the two EnKF scenarios. The first, second, and third rows correspond to the node-based, lens-based, and meander-based scenarios. Columns one to four illustrate results of the standard EnKF scenarios, columns five to eight those of the GA-EnKF scenarios. The first and fifth columns show individual parameter realizations at the end of the assimilation period, the columns to their right the corresponding ensemble average. The third and seventh columns illustrate the final standard deviation of the hydraulic

conductivities, and the fourth and eighth column the standard deviations of the hydraulic heads at the end of the assimilation period.

Inspecting the optimized parameter fields of the EnKF scenarios, we observe that both scenarios have identified the general structure of the reference conductivity field. Unfortunately, however, we also note that the realizations of the final ensemble (Figure 2-6, columns 1 and 5) have all but lost the geological features defined in the initial field generation. In the standard EnKF, remnants of the initial structural features still remain (Figure 2-6, column 1), whereas the GA-EnKF seems to have completely erased the structural differences. The standard deviations of parameter uncertainty (Figure 2-6, columns 3 and 7) for the lens-based and meander-based hyperparameterizations (Figure 2-6, rows 2 and 3) further indicate regions in the south-eastern quadrant for which the ensemble seemed to have collapsed against the prescribed lower bound of the hydraulic conductivity, in turn also collapsing the local state uncertainties (Figure 2-6, columns 4 and 8). This effect is more pronounced for the standard EnKF than for the GA-EnKF scenario. Finally, we note that for both EnKF scenarios the node-based hyperparameterization appears to retain the highest state and parameter uncertainty at the end of the data assimilation period.

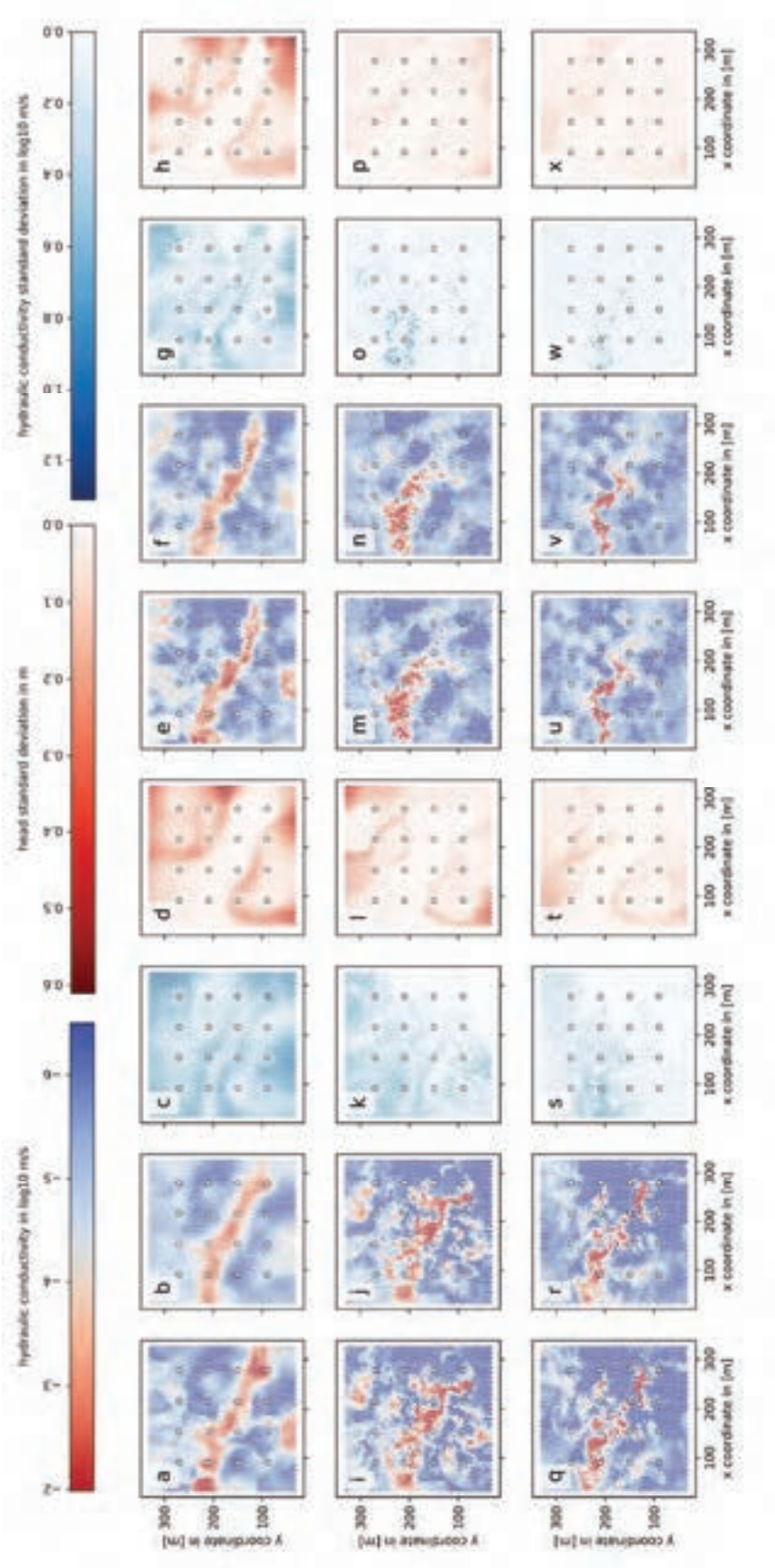


Figure 2-6. Selected parameter fields (columns 1, 2, 5, 6) and the standard deviations of conductivities (columns 3 and 7) and heads (columns 4 and 8) obtained by the EnKF for the three field generators at the end of the data assimilation period: node-based (row 1), lens-based (row 2) and meander-based (row 3); the first column (a, i, q) illustrates selected realizations of the standard EnKF scenario; the second column (b, j, r) shows the ensemble mean of the standard EnKF scenario; the third (c, k, s) and fourth column (d, l, t) illustrate the standard deviations of parameters and heads; the fifth (e, m, u), sixth (f, n, v), seventh (g, o, w), and eighth (h, p, x) are the corresponding entries of the GA-EnKF scenario.

2.5.2 Parameter estimation performance

We now analyse the optimized parameter fields and their resulting state predictions. The perfect knowledge of the synthetic model allows us to investigate the deviations from the reference even at locations where otherwise no information would be available. In order to avoid over-interpreting statistical artefacts, we investigate the parameter and state fields averaged over all ten random seeds. Figure 2-7 illustrates the state deviations (Figure 2-7a, c, e) and the parameter deviations (Figure 2-7b, d, f) for the node-based (Figure 2-7a, b), lens-based (Figure 2-7c, d), and meander-based (Figure 2-7e, f) scenarios. The corresponding standard deviations are illustrated in Figure I-9, their counterparts for the two EnKF scenarios are provided in Figure I-10 to Figure I-13, and deviations for the hybrid nested particle filter are illustrated in Figure I-14 (supporting information).

Comparing the state deviations of the nested particle filters (default and hybrid setups), we note a few interesting aspects: All three scenarios managed to reproduce the state observations along the synthetic meander relatively well. This is a result of the high correlation between observed heads within the meander and results in a heavy likelihood penalty should this continuous, high-conductive structure be broken or not created. We further note that on average both the node-based and lens-based scenarios identify conductivity fields which reproduce head observations at the observation points well but deviate from the reference between measurement points. It is worth noting that the node-based hyperparameterization develops larger state deviations in-between observation points than the lens-based case. A possible explanation is that the node-based approach allows an almost continuous adjustment of hydraulic conductivity, whereas the lens-

based method is essentially binary. The meander-based case succeeds in reproducing observations within the meander but deviates substantially for outlying observation wells.

To explain this inability to reproduce the state observations farther into the background facies, it may help to consider the parameter field deviations in the light of the mischaracterization of heterogeneity (Figure 2-4). While the optimization algorithm identifies the correct mean background conductivity, the state mismatch suggests that this mischaracterization results in an under-prediction of states compared to the reference. This phenomenon is likely rooted in an inability to replicate the effects of the reference's internal heterogeneity patterns. The lens-based scenario is subject to the same fundamental mischaracterization but the independent placement of high-conductive lenses at strategic locations within the background facies allows compensating this structural error. A closer look at the internal background heterogeneity reveals that the optimizer places clusters of high-conductive lenses along the north-east or south-west of the model domain in the node-based and lens-based scenarios. We believe that these placements compensate the unresolved internal variability of the background field. In the node-based scenario, we find parameter fields with features similar to the lens-based case, despite much greater freedom in the distribution of conductivities. Recurring features include high-conductive clusters in the south-west, an underestimation of conductivity between the second, third, sixth, and seventh observation wells (counting left to right, top to bottom) or in the south-east. These recurrences suggest that a number of features of the reference conductivity field may not be sufficiently informed by the head

observations, and that there likely are more stable, or more easily identifiable, equivalent solutions to the reference.

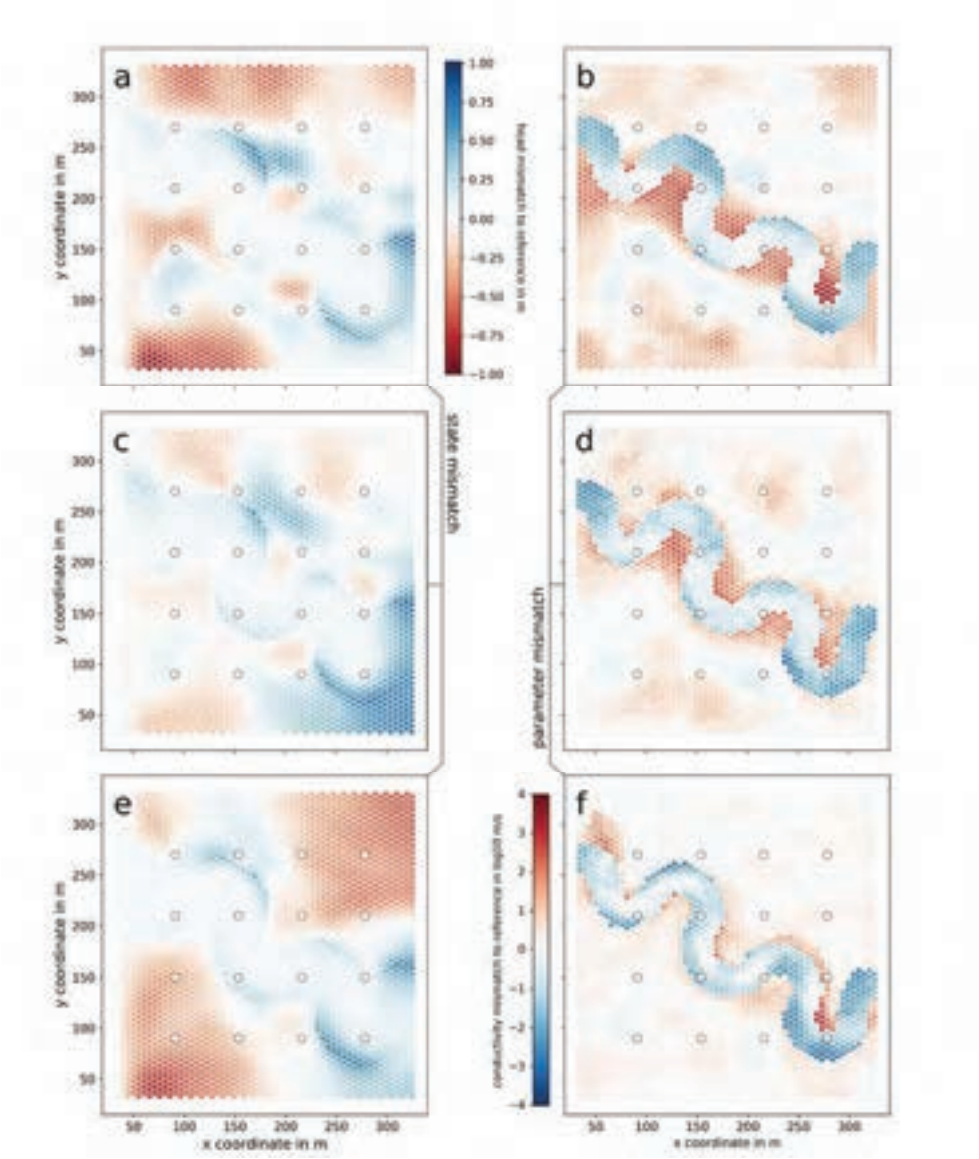


Figure 2-7. Mismatch between the synthetic reference and ensemble mean at the end of the data assimilation period for hydraulic heads (a, c, e) and hydraulic conductivities (b, d, f), averaged over ten random seeds for the nested particle filter scenarios. Results are shown for the node-based (a, b), lens-based (c, d), and meander-based (e, f) hyperparameterizations. Mind that the relation of colors to quantities is reversed between the state mismatch (a, c, e) and the parameter mismatch (b, d, f) columns. This was a deliberate choice to visually underline the common relation of parameter underestimation to state overestimation, and vice versa.

2.5.3 Predictive performance

As the gradual improvement of model predictions is a major objective of the nested particle filter, it may be worth investigating prediction performance without data assimilation and the explicit assumption of

time-varying parameters. To do so, we extract the expected parameter values at the end of each cycle and repeat a deterministic simulation over 90 recharge cycles for both the particle filter and EnKF scenarios. After a dynamic steady state has been established, we determine the average root-mean square error between the states $\boldsymbol{x}_{predicted}$ and the corresponding synthetic reference $\boldsymbol{x}_{synthetic}$ at the observation points over the final recharge period $s = 1, \dots, 100$. We refer to this quantity as the prediction root-mean square error (pRMSE) and calculate it according to:

$$pRMSE = \frac{1}{N_{obs}} \sum_{o=1}^{N_{obs}} \sqrt{\frac{1}{100} \sum_{s=1}^{100} \left(x_{predicted}^{(o,s)} - x_{synthetic}^{(o,s)} \right)^2} \qquad 2\text{-}17$$

We furthermore calculate the percentage bias of the predicted states according to:

$$pbias = \frac{1}{N_{obs}} \sum_{o=1}^{N_{obs}} \sum_{s=1}^{100} \frac{x_{predicted}^{(o,s)} - x_{synthetic}^{(o,s)}}{x_{synthetic}^{(o,s)}} \qquad 2\text{-}18$$

The results of this evaluation are visualized in Figure 2-8.On average, we observe an increase in predictive performance across all hyperparameterization scenarios, albeit with varying degrees of stability and fidelity. For the nested particle filter, the final pRMSEs of the node-based, lens-based and meander-based scenarios fall approximately between 0.05 and 0.2 m, 0.1 and 0.4 m, and 0.15 and 0.3 m, respectively. The final relative biases of the predictions for the node-based, lens-based and meander-based scenarios fall between +/-10%, -10% to +20%, and +/-10%, respectively. Initially, all three scenarios display – for most random seeds – a positive bias but vary in the magnitude of their respective pRMSEs. The discrepancies in initial pRMSEs can be explained by the

nature of the different hyperparameterizations, and how much they are constrained by prior geological information. The node-based scenario is barely constrained, with the contribution of the three nodes of known conductivity lost among the effect of 50 randomized nodes. The lens-based scenario featured the highest prior pRMSEs, as a replication of the reference's state distribution requires a specific arrangement of lenses unlikely to emerge by chance. The meander-based case displayed the lowest prior pRMSEs, owing to the strong constraint placed on the meander's path – it has to pass through a specific well in the north-west – and the overall low number of hyperparameters, increasing the chances of creating a prior parameter particle near an optimum. It is worth noting that we are free to adapt the ensemble size during the optimization process. This would, for example, allow us to initialize the filter with a large ensemble of prior particles, identify the best among them through weighting, and then resample (and proceed with) a smaller ensemble.

Aside from long-term optimization, however, we observe short-term deviations from prevailing optima. The occurrence of such instabilities is not surprising given the probabilistic nature of the artificial parameter dynamics but is exacerbated by the limited cycle length. Across the scenarios, we observe a qualitative drop in optimization stability from the node-based, through the lens-based, to the meander-based hyperparameterization. The relative stability of the node-based scenario is based in the absence of hyperparameters with a large-scale impact on the conductivity field, effectively limiting the 'damage' from resampling a sub-par mutation. Conversely, the instability of the meander-based scenario (observed in both the default and hybrid setup) can be explained by the strong interaction between its hyperparameters, none

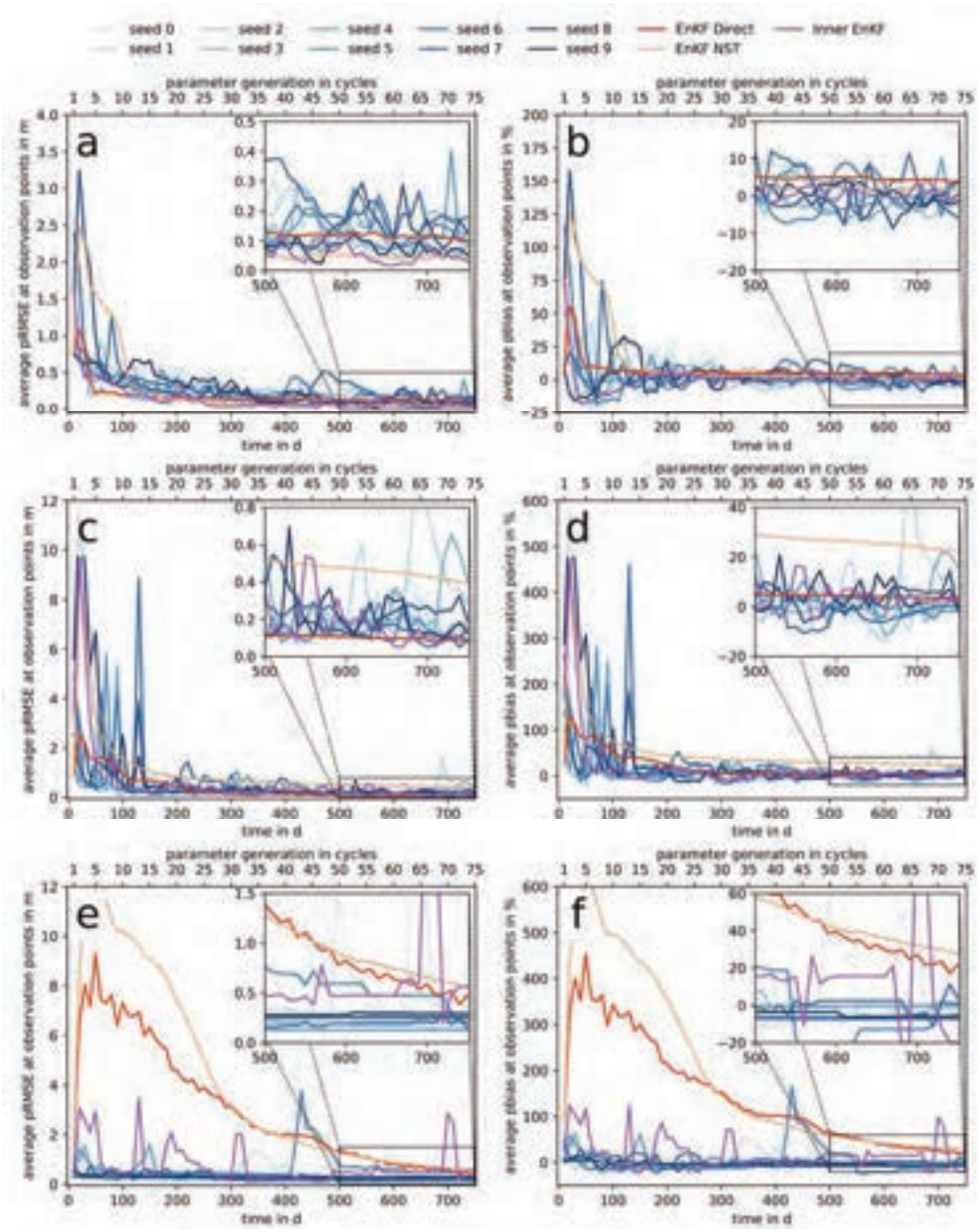


Figure 2-8. Development of pRMSE (a, c, e) and percentage bias (b, d, f) through time for the node-based (a, b), lens-based (c, d) and meander-based (e, f) scenarios and ten different random seeds, the Inner EnKF scenario, and the two EnKF scenarios.

of which can be altered without large-scale consequences. This also provides an explanation for the apparent presence of local optima, which become evident by prolonged sequences during which proposed mutations are repeatedly rejected (e.g., Figure 2-8 c, random seed 5). The lens-based scenario appears to be as stable as the node-based one, but occasionally expresses larger deviations, most likely because some of its hyperparameters have a larger impact on the resulting parameter field.

We note that the magnitude of this instability could be alleviated by reducing the strength of the mutations over time, a common practice in methods like simulated annealing. In general, the hybrid scenarios with inner EnKFs yield similar performance to the other nested particle filter setups.

For the full EnKF scenarios, optimization performance was generally more stable. Both standard EnKF and GA-EnKF showed similar performance, although the standard EnKF seemingly optimized quicker. More significant deviations in performance were found between the different geological scenarios. Since the EnKF scenarios only uses the field generators during the generation of the prior ensemble and then updates the ensembles as if they were multi-Gaussian, the observed differences must originate from variations in the properties of the prior covariance matrices. Extracting the eigenvalues of these matrices (Figure I-15), we note that the node-based covariance matrix has the highest, the meander-based case the lowest, and the lens-based case an intermediate effective dimensionality. For the node based scenarios – offering the largest initial parameter uncertainty with the least spatial correlation – optimization performance was quick and precise, matching or surpassing the nested particle filters. For the lens-based hyperparameterization, the standard EnKF appears to out-perform the GA-EnKF in both pRMSE and relative bias and matches the best of the nested particle filters. The GA-EnKF performs worse than all nested particle filters, although Figure 2-8c and e suggest steady improvement until the end. This is likely owed to filter collapse, as Figure 2-6f and Figure 2-6h show that the parameter uncertainty collapsed for both scenarios towards the end. A similar collapse seemingly occurred for the meander-based hyperparameterization (Figure 2-6j and l): both

EnKF scenarios feature worse pRMSEs and relative biases than the nested particle filters, although steady improvements are observed until the end. This ensemble collapse could be remedied with active covariance inflation.

2.5.4 Discussion

The results obtained suggest the nested particle filter can optimize even highly complex hyperparameterizations, but different degrees of optimization performance divide the investigated scenarios in three classes: an over-flexible setup (node-based), a balanced setup (lens-based), and an over-constrained setup (meander-based).

Despite given the correct geological structure, the meander-based scenario performed poorest. Unable to compensate for a fundamental mischaracterization of the background facies heterogeneity, we found that the hyperparameterization could not generate parameter fields adequately replicating the synthetic reference's hydraulic head distribution. Furthermore, high dependence among the hyperparameters suggests the presence of numerous local optima and unstable optimization performance.

Despite featuring an erroneous geology and being subject to the same fundamental mischaracterization of heterogeneity, the lens-based hyperparameterization performed favourably. Its optimized parameter fields revealed that the lenses can arrange themselves to form features functionally similar to the reference's high-conductive meander and even compensated for the mischaracterized heterogeneity. While the prediction performance did not fully measure up to the node-based scenario, its geological characterization resulted in smaller deviations

from the reference at points where no information would otherwise be available.

Unencumbered by constraints of complex geology and with many virtually independent hyperparameters, the node-based scenario permitted the most flexible adjustment of the conductivity field. While this scenario best replicated the reference's state observations, a comparison to the synthetic reference's latent states revealed substantial deviation in-between observation wells. This suggests that the quality of the fit was at least partially afforded by over-parameterization.

Results for the hybrid nested particle filter with inner EnKFs were similar to those obtained with the normal nested particle filters, owed to an irreversible collapse of state uncertainty which eliminated differences between the two setups early on. In a real setting with non-negligible forecast errors and uncertain forcing, we expect that these sources of entropy would retain more uncertainty and permit the hybrid filter to better leverage the efficiency of its inner EnKF.

For the full EnKF reference scenarios, we note that the best performance was achieved for the node-based scenario, closely followed by the lens-based scenario. The lens-based and the meander-based scenarios both suffered from ensemble collapse. This suggests that the EnKF optimization performs best if the initial ensemble encloses a large volume of parameter space and may perform poorly in scenarios with larger spatial correlations (lens-based and meander-based). Where the ensembles did not collapse, the EnKF can yield parameter fields on par to or better than those from the nested particle filters. Unfortunately, the EnKF's update procedure abandons the support of the geological prior in the process: the EnKF did not sufficiently honor the initial geological

features in all scenarios considered, yielding posterior parameter fields strongly deviating from the prescribed geology. This suggests that if conformance with a geological prior is of the essence, the use of data-assimilation and calibration schemes which implicitly rely smoothness or regularity assumptions in conflict with the prior's support (e.g., Gaussianity for the EnKF) may not be advisable. In the present study, we suggest geology-obeying hyperparameterizations, but MPS with geologically realistic training images may also work.

Summarizing the results, we find that the nested particle filter can successfully optimize hyperparameters whose relation to the state response is highly non-linear or discontinuous. The scenarios investigated in this study revealed a number of important aspects: While conformance to arbitrary geology can be enforced with ease, it is important to leave the algorithm sufficient flexibility to compensate for potential (and in practice inevitable) mischaracterizations of geology. In the case of the meander-based field generator, further hyperparameters adjusting the mischaracterized heterogeneity might have improved the performance. A general way to achieve this is by providing options to locally adjust parameter fields, as shown by the lens-based and node-based scenarios. Acknowledging that this compensation for structurally wrong parameters (*parameter surrogacy*: John Doherty & Christensen, 2011) is often undesirable, its occurrence may nonetheless reveal fundamental conceptual errors.

2.6 Conclusions

In this study, we explored the use of a nested particle filter framework, a generalization of the classic particle filter for joint state-parameter estimation, for real-time parameter optimization in distributed

groundwater models. We made use of hyperparameterized field generators to reduce the dimensionality of the optimization problem and to guarantee conformance to a prescribed geology throughout the optimization process, using variance inflation through artificial parameter dynamics to rejuvenate the parameter particles. Examining the performance for three simple field generators, we identified versatile hyperparameterization as a prerequisite for the algorithm's success in mischaracterized settings.

We then compared the optimizer to two classic EnKF setups – a standard state-vector augmented EnKF, and a state-vector augmented EnKF with Gaussian anamorphosis, both initialized with samples from the different field generators. While their optimization performance can be equal or even superior with sufficient initial variability in the parameter ensemble, none of the scenarios considered sufficiently preserved the prescribed geological structures. This is a consequence of the EnKF updating its ensembles as if they were multi-Gaussian in combination with the highly non-Gaussian support of the parameter priors in grid parameter space. Where geological fidelity is essential, we thus suggest to combine field generators (e.g., MPS, object-based generators) with optimization routines capable of reliably navigating the prior's support. As such, our results make us believe that the nested particle filter – if adequately (hyper)parameterized – could constitute a valuable complement to other real-time parameter estimation methods, particularly in scenarios where the conservation of complex geological features is critical.

We would like to remark that it would theoretically also be possible to have an EnKF operate on a joint state-hyperparameter space. This would likely impose a number of restrictions on the design of the field generator

to ensure sufficient regularity, and possibly require post-analysis sanity-checks to ensure consistency with geological conditioning data. It is furthermore to be expected that the linear relation between state response and hyperparameters would be weakened. The design of such EnKF-friendly hyperparameterizations was beyond the scope of the present study. It remains open to what degree the efficiency of the EnKF justifies these self-imposed restrictions, but we note that this could be an interesting avenue for future research. Hybrid solutions such as the inner EnKF scenarios considered in this study are also a promising way to leverage the EnKF's efficiency for the state updates while retaining the particle filter's generality for the parameter updates.

A limitation of this study is the lack of uncertainty retained in the particle ensemble, a consequence of the filter's ensemble collapse, although the framework is in principle capable of sustaining such information and may do so in different settings or with different likelihood functions. A further limitation is the use of artificial parameter dynamics, which may preclude an application in systems with longer memory if physical consistency between states and parameters is of concern. Continuing research could address these issues by adapting the cycle length across the optimization period, adjusting mutation magnitude and initializing the filter with a larger parameter ensemble size. Applications in less dissipative settings, such as the simulation of remediation efforts, could be approached by employing a nested particle filter with a different rejuvenation mechanism such as SMC^2. The arbitrary nature of parameter dynamics also provides an adaptive interface to other field generators, for example MPS. The flexibility of the nested particle filter framework encourages experimentation with different numerical models, parameter dynamics, states, and (hyper)parameters. The

algorithm presented herein extends trivially to 3-D, as none of its elements places a restriction on the spatial dimensionality of the numerical model.

2.7 Acknowledgments

We would like to thank Dr. Christian Langevin of the U.S. Geological Service for his support in implementing the Python-MODFLOW interface FloPy around which this algorithm was developed and tested. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675120. The data necessary to reproduce the results of this paper are available under this DOI: https://doi.org/10.25678/000161.

2.8 Appendix 1: Investigation of the ensemble collapse

In Section 2.5.1 we noted that the outer particle filter in this study collapses during each parameter cycle. In pursuit of uncertainty information pertaining the parameter distribution, this is of course undesirable. In this appendix, we investigate the ensemble collapse with a back-of-the-envelope calculation to identify the reasons for this persistent collapse, and how the framework would have to be parameterized to allow for a better conservation of parameter uncertainty.

As we established in Section 2.3.2, the ensemble collapses if only one particle retains any significant non-zero weight. According to Equation 2-14, the parameter particle weights are proportional to the

marginal likelihood derived over the arithmetic mean of its inner particle filter's likelihoods. We summarize:

$$w_c^{(n_\theta)} \propto \mathcal{L}_c^{(n_\theta)} = \frac{1}{N_x} \sum_{n_x=1}^{N_x} \prod_{u=1}^{L} \prod_{o=1}^{N_{obs}} l_z^{(n_\theta, n_x, n_o)} \qquad 2\text{-}19$$

Since we want to learn about the relations required to prevent degeneracy, it may be useful to consider a simplified case in which $l_z^{(n_\theta,n_x,n_o)} = \overline{l^{(n_\theta)}},\ \forall\, n_x, n_o, u$, which leads to the following simplification:

$$w_c^{(n_\theta)} \propto \frac{1}{N_x} \sum_{n_x=1}^{N_x} \prod_{u=1}^{L} \prod_{o=1}^{N_{obs}} l_z^{(n_\theta, n_x, n_o)} = \overline{l^{(a)}}^{LN_{obs}} \qquad 2\text{-}20$$

The prevention of the parameter filter collapse requires that at least some parameter particles retain a non-zero weight ratio. Omitting the time subscripts from the notation for the sake of simplicity, we define a weight ratio r:

$$r = \frac{w^{(b)}}{w^{(a)}}, \qquad 0 \le r \le 1, \qquad w^{(b)} \le w^{(a)} \qquad 2\text{-}21$$

where $a, b \in 1, \dots, N_\theta$. Using the simplification introduced in Equation 2-20, we can adapt the expression for the likelihood Equation 2-9:

$$\overline{l^{(n_\theta)}} = \frac{1}{\sqrt{2\pi\sigma_{obs}^2}} e^{\left(-\frac{\left(\overline{\Delta^{(n_\theta)}}\right)^2}{2\sigma_{obs}^2} \right)} \qquad 2\text{-}22$$

where $\overline{\Delta^{(n_\theta)}}$ denotes the mismatch between state prediction and observation, which – in accordance with the simplification introduced in Equation 2-20 – is assumed equal for all state particles, observations, and time steps for the sake of this exercise. Combining Equation 2-20, Equation 2-21 and Equation 2-22, we can derive an expression for the

weight ratio depending on the observation mismatches between two parameter particles a and b:

$$R = e^{\left(LN_{obs}\frac{\left(\Delta^{(a)}\right)^2}{2\sigma_{obs}^2} - LN_{obs}\frac{\left(\Delta^{(b)}\right)^2}{2\sigma_{obs}^2}\right)} \qquad 2\text{-}23$$

which can be re-formulated to yield an expression for the required relation between observation mismatches given a weight ratio R:

$$\frac{2\sigma_{obs}^2 \ln R}{LN_{obs}} = \left(\Delta^{(a)}\right)^2 - \left(\Delta^{(b)}\right)^2 \qquad 2\text{-}24$$

As R is bounded between 0 and 1 and all other variables are positive, the term on the left-hand side is either zero or negative. Since we further defined $w^{(b)} \leq w^{(a)}$, we get $\Delta^{(a)} \leq \Delta^{(b)}$. We can reformulate Equation 2-24 into an expression for $\Delta^{(b)}$ given $\Delta^{(a)}$:

$$\Delta^{(b)} = \sqrt{(\Delta^{(a)})^2 - \frac{2\sigma_{obs}^2 \ln R}{LN_{obs}}} \qquad 2\text{-}25$$

Results for the values used in this study ($L = 10$, $N_{obs} = 16$, $\sigma_{obs} = 0.02$) are illustrated in Figure 2-9 for different weight ratios. The larger a given particle's average observation mismatch $\overline{\Delta^{(a)}}$, the less a second particle's average observation mismatch $\overline{\Delta^{(b)}}$ is allowed to deviate from it to conform to the desired weight ratio. The order of magnitude of δ reveals the reason for the observed ensemble collapse. Assuming we deem a weight ratio of 10 % sufficient, and assuming one particle has an average observation mismatch of 5 cm, a second (inferior) parameter particle could deviate at most 1.15 µm from the average observation mismatch before causing degeneracy – a highly improbable event.

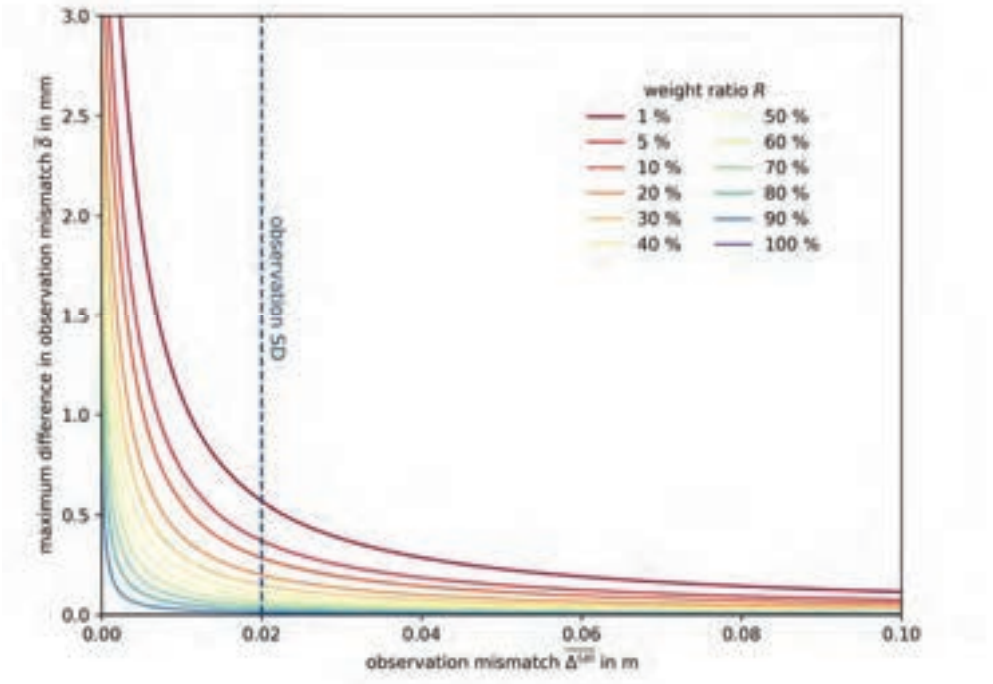


Figure 2-9. Relation between average observation mismatch and the maximum observation mismatch discrepancy to a second particle for several weight ratios, derived under the assumption $|\overline{\Delta^{(b)}}| > |\overline{\Delta^{(a)}}|$. Parameters used to obtain these results correspond to the ones used in this study: $\boldsymbol{L = 10}$, $\boldsymbol{N_{obs} = 16}$, $\boldsymbol{\sigma_{obs} = 0.02}$.

Based on this simplified representation, we can determine a number of variables that can help prevent weight degeneracy: Shortening the cycle length L is a possibility, but risks carry-over effects arising from insufficient dissipation. Decreasing N_{obs} or increasing σ_{obs}^2 could also yield improvements, but both variables are generally out of the user's hand. A more viable solution strategy may be to reduce the magnitude of changes proposed in the artificial parameter dynamics to reduce the resulting deviation in state predictions.

3 Quasi-online groundwater model optimization under constraints of geological consistency based on iterative importance sampling[13]

3.1 Abstract

The increasing use of wireless sensor networks and remote sensing permits real-time access to environmental observations. Data assimilation frameworks tap into such data streams to autonomously update and gradually improve numerical models. In hydrogeology, such methods are relevant in areas of long-term interest in water quality and quantity, for example in drinking water production. Unfortunately, accurate hydrogeological predictions often demand a degree of geological realism which is difficult to reconcile with the operational limitations of many data assimilation frameworks. Alluvial aquifers, for example, are sometimes characterized by paleo-channels of unknown extent and properties which may act as preferential flow paths. Gradually optimizing such fields in real-time or quasi-real-time settings is a formidable task. Besides subsurface properties, ill-specified model forcings are a further source of predictive bias which an optimizer could learn to compensate. In this study, we explore the use of a quasi-online optimizer based on the iterative batch importance sampling (IBIS) framework for a groundwater model of a field site near Valdobbiadene, Italy. This site is characterized by the presence of paleo-channels and

[13] This chapter has been published in Water Resources Research: Ramgraber, M., Camporese, M., Renard, P., Salandin, P., & Schirmer, M. (2020). Quasi-online groundwater model optimization under constraints of geological consistency based on iterative importance sampling. Water Resources Research. https://doi.org/10.1029/2019wr026777.

heavily exploited for drinking water production and irrigation. We use Markov Chain Monte Carlo (MCMC) steps to explore new parameterizations while maintaining consistency between states and parameters as well as conformance to a multi-point statistics (MPS) training image. We also optimize a pre-processor designed to compensate for potential bias in the model forcing. We achieve promising and geologically consistent quasi-real-time optimization, albeit at the loss of parameter uncertainty.

3.2 Introduction

Groundwater is a critical resource for sustainable human and ecological development (Burri et al., 2019; Schirmer et al., 2013), constituting the dominant source of drinking water and irrigation in many countries across the globe (e.g., Gorelick & Zheng, 2015). As such, safeguarding this resource is a long-term endeavor across multiple time scales, ranging from months/years (e.g., dewatering of contruction sites: Powers et al., 2007), to decades (e.g., drinking water production, aquifer remediation: Kresic & Stevanovic, 2010; UNICEF, 2016) or even centuries (e.g., prevention of saltwater intrusion: Oude Essink, 2001; A. D. Werner et al., 2013).

Numerical modeling plays a crucial role in informing such hydrogeological practices (Reilly & Harbaugh, 2004). The parameterization of groundwater models demands a full characterization of subsurface properties, information which can only partially be obtained from direct measurements. Consequently, modelers often find themselves tasked with the synthesis of plausible parameter fields from different sources of information (McLaughlin & Townley, 1996; Yeh, 1986) such as geological field characterizations (e.g.,

Linde et al., 2015; Zovi et al., 2017), geophysical measurements (e.g., Zovi et al., 2017), or parameter-dependent state observations (e.g., John Doherty et al., 2010; Sun, 1994).

As the most accessible data type in inverse modelling, state observations can provide a wealth of information. Unfortunately, depending on climatic conditions, they might reveal only certain aspects of the subsurface at any given time. As such, the process of assembling a sufficiently informative data set may be slow. This motivates the use of online model optimization routines based on data assimilation (e.g., Hendricks Franssen & Kinzelbach, 2008). While the adoption of these methods into practice is still in its infancy, such algorithms could connect to online sensor networks and assimilate data as it becomes available, sequentially optimizing a model's state and parameter estimates. In hydrogeology, certainly the most popular among such techniques is the *Ensemble Kalman Filter* (*EnKF*: Evensen, 1994, 2003). This method has seen a great rise in popularity over the past decades (e.g., Aanonsen et al., 2009; Hendricks Franssen & Kinzelbach, 2008; Reichle et al., 2002; Tang et al., 2015, 2017; Zhou et al., 2011) owed to its simplicity, relative ease of implementation, great computational efficiency, and remarkable robustness to both small ensemble sizes and violations of its implicit assumptions of *linearity* and *Gaussianity*.

However, the price for the EnKF's elegance is that it updates its variables *as if* its fundamental assumptions were met (e.g., Katzfuss et al., 2016), which is rarely – if ever – the case in hydrogeology. Particularly complex geological priors often deviate substantially from Gaussianity (Aanonsen et al., 2009; A. Y. Sun et al., 2009). As such, the common practice of optimizing log-conductivity fields sampled from such priors with the EnKF (e.g., Jafarpour & McLaughlin, 2009; Tang et al., 2015,

2017; Zhou et al., 2011; Zovi et al., 2017) risks leaving the support of the prior. This, in turn, means that the EnKF eventually erases geological features present in the initial ensemble (Ramgraber et al., 2019; Zovi et al., 2017), and yields posterior samples incompatible with the prior. Attempts to enforce conformance by construction (e.g., Hu et al., 2013) circumvent this issue, but instead often suffer from a weakened linear relation between parameter changes and state response, exploited by the EnKF's parameter update (e.g., Crestani et al., 2013).

In order to achieve a good fit to hydraulic heads, practitioners often neglect geological realism or structural uncertainty in favor of simpler formulations such as a-priori zonation with homogeneous properties or interpolation from a set of pilot points (Cirpka & Valocchi, 2016). While this may prove adequate for the prediction of flow only, using a model for transport-related quantities (e.g., flow paths, travel times, reactive transport) demands a more faithful representation of the geology (Alcolea & Renard, 2010; Cirpka & Valocchi, 2016; Fogg & Zhang, 2016; Sanchez-Vila & Fernàndez-Garcia, 2016).

To reconcile the challenges of geological realism with the operational limitations of sequential optimization frameworks, it seems auspicious to return to more general sequential Monte Carlo (SMC) techniques like particle filters (e.g., Doucet & Johansen, 2009; Doucet & Tadić, 2003; van Leeuwen, 2009; van Leeuwen et al., 2019). These methods promise greater freedom in exploring complex probability distributions with nonlinear relations between parameters and states. The price for this flexibility is often drastically lower efficiency: classic particle filters demand an ensemble size exponential with regard to the dimensionality of the system (e.g., Snyder et al., 2015), a restriction known as the *curse of dimensionality*. Failing to provide a sufficiently large ensemble– a virtual

inevitability given the high dimensionality and computational cost of most subsurface models –results in *sample degeneracy* (e.g., Doucet & Johansen, 2009; Li et al., 2015) and eventually the collapse of the particle approximation. While this degeneration of the particle approximation often cannot be avoided in practice, it can still provide a powerful basis for model optimization.

In this study, we construct a quasi-online optimizer based on *Iterated Batch Importance Sampling* (*IBIS*: Chopin, 2002; Chopin et al., 2013), a particle filter that uses Markov Chain Monte Carlo (MCMC) steps to counteract sample degeneracy (*rejuvenation*) and optimizes the ensemble in the process. While MCMC steps are a common rejuvenation mechanism in hydrological particle filters (e.g., Moradkhani et al., 2012; Noh et al., 2011; Vrugt et al., 2013), many methods simplify the point-wise evaluations of the posterior probability density it requires by using intermediate density estimates from the particle approximation. In the IBIS algorithm – similarly to the Restart EnKF (Gu & Oliver, 2007) –, the full observation history is re-simulated instead and the posterior density is computed directly. This guarantees that states and parameters are always internally physically consistent and renders the fidelity of the rejuvenation mechanism largely independent of the (possibly degenerate) particle approximation, at the cost of steadily increasing computational demand. In our study, we compensate this effect by dynamically adjusting the ensemble size, and use the flexibility of the MCMC framework to sequentially optimize a model under a complex geological prior, maintaining conformance by construction through a combination of hyperparameterization and multi-point statistics (*MPS*: Caers et al., 2003; Journel & Zhang, 2006). The optimizer is tested at a field site in northern Italy characterized by paleo-channels, the object of

a previous study employing the EnKF (Zovi et al., 2017). We implement the algorithm in three different scenarios and compare the results obtained to the previous study.

3.3 Theory

3.3.1 Nomenclature

In probabilistic systems model variables are separated into two classes. The *parameters* θ are usually static model variables, such as hydraulic conductivities or specific yield, and generally independent from other variables. *States*, denoted by x, are typically time-varying quantities which depend on parameters or model forcing. Hydraulic heads, temperatures, or concentrations all are common examples. The *observations* y – generally measurements of states – are treated as a third, separate variable type.

All system variables of the same type are combined into a vector and interpreted as coordinates of a point in high-dimensional variable space (*parameter space, state space*, and *observation space*, respectively). *Particles* occupy one such point and thus represent a full set of the respective variable type required by the model. We assign a superscript index in brackets to individual particles and their associated variables, e.g. $x^{(index)}$. Time-dependent variables are designated by subscripts x_{time} for specific time points, and time spans between a start and end point are represented by $x_{start:end}$. A '~' should be read as 'sampled from' and a semi-colon ';' denotes 'parameterized by'. Figures and tables numbered with a leading 'S' refer to material in the supporting information. A list of all variables is provided in Table S1.

3.3.2 Sequential Bayesian inference

At the heart of Bayesian parameter estimation lies the inference of the posterior *probability density function* (pdf) $p(\theta\,|y_{1:t})$, which can be determined to proportionality through the prior $p(\theta)$ and the likelihood of the observation time series conditional on the parameters $p(y_{1:t}|\theta)$:

$$p(\theta\,|y_{1:t}) \propto p(\theta)p(y_{1:t}|\theta) \qquad 3\text{-}1$$

Sequential data assimilation frameworks incorporate data in increments. Assuming time-independent likelihoods, we can reformulate 3-1 to obtain:

$$p(\theta\,|y_{1:t}) \propto p(\theta)\prod_{s=1}^{t} p(y_s|\theta) \qquad 3\text{-}2$$

While $p(\theta)$ is generally user-prescribed, the likelihood $p(y_s|\theta)$ is not always straightforward to obtain. Most EnKF variations and particle filters based on state-vector augmentation (e.g., Moradkhani et al., 2005) instead use the observational likelihood $p(y_s|x_s)$, then extend the Bayesian update to the parameters via the parameter-dependent states. Nested particle filters (e.g., SMC^2: Chopin et al., 2013) also use the observational likelihood, but then integrate over the state space to obtain $p(y_s|\theta)$.

An alternative approach is to omit the model states entirely from the probabilistic part of the inference process. Since most numerical groundwater models $M(x_0, u_{1:t}, \theta)$ are deterministic to begin with, one may interpret the model states $x_{1:t}$ as the output of a deterministic mapping from parameter space, the initial states x_0 and the model forcing $u_{1:t}$. If we further assume that x_0 and $u_{1:t}$ depend only on θ and constants, this dependency effectively reduces to θ, and we can consider x_0 and $u_{1:t}$ intermediate results of the deterministic map from θ to $x_{1:t}$.

The states $x_{1:t}$, in turn, map deterministically to observation space, providing us with a deterministic map of parameters to observations by 'bridging across' state space. This process is often called a *forward operator* (also: *forward solver*, *forward map*: Linde et al., 2015; McLaughlin & Townley, 1996):

$$\theta \overset{det.}{\longrightarrow} x_{1:t} \overset{det.}{\longrightarrow} y_{1:t}^{sim} \tag{3-3}$$

The map from state space to observation space ($x_{1:t} \overset{det.}{\longrightarrow} y_{1:t}^{sim}$) can be a non-linear function (e.g., van Leeuwen, 2015), but in hydrogeology – where the observed quantities are generally simulated directly – it can often be simplified to a dot product with a matrix H extracting the relevant entries from the state vector:

$$y_s^{sim} = Hx_s \tag{3-4}$$

where H is a matrix of zeros and ones. Combining 3-3 and 3-4 yields

$$\theta \overset{det.}{\longrightarrow} x_{1:t} = M(x_0, u_{1:t}, \theta) \overset{det.}{\longrightarrow} y_{1:t}^{sim} = \begin{bmatrix} y_0^{sim} \\ \vdots \\ y_t^{sim} \end{bmatrix} = \begin{bmatrix} Hx_0 \\ \vdots \\ Hx_t \end{bmatrix} \tag{3-5}$$

Given this deterministic projection into observation space we require an error model to permit a probabilistic analysis of the observed data. One possible way to do so is through the addition of a lumped, additive, multivariate Gaussian error centered around the forward operator's output $y_{1:t}^{sim}$ with a specified covariance matrix Σ. Since we assume spatially and temporally uncorrelated errors in our study (3-2), the error model could theoretically be broken down into a product of univariate Gaussians. To better reflect the incremental structure in which new observations become available, we instead define it as a multivariate Gaussian over all observation points available at a given timestep s

centered on the predicted observations y_s^{sim} with a diagonal covariance matrix Σ:

$$p(y_s|\theta) = \mathcal{N}(y_s; \mu = y_s^{sim}, \Sigma) = \frac{1}{\sqrt{(2\pi)^{N_s^{obs}} \det \Sigma}} \exp\left(-\frac{1}{2}(y_s - y_s^{sim})^\top \Sigma^{-1}(y_s - y_s^{sim})\right) \quad 3\text{-}6$$

where N_s^{obs} denotes the number of elements in the observation vector y_s. With the likelihood $p(y_s|\theta)$ defined, we can proceed to the specific algorithm used in this study.

3.3.3 Iterated Batch Importance Sampling (IBIS)

The Iterated Batch Importance Sampling (IBIS) algorithm was introduced by Chopin (2002) for the sequential filtering of static parameters, provided that the likelihood $p(y_s|\theta)$ can be evaluated. Since Bayes theorem (3-1) is all but impossible to solve analytically in the general case, it often becomes necessary to resort to Monte Carlo approximations. These methods assume that a set of Monte Carlo samples (an *ensemble* of *particles*) may act as a surrogate for the distribution from which they were drawn. *Sequential Monte Carlo* (*SMC*) methods, then, try to retain this surrogate property along otherwise intractable Bayesian update operations through gradual updates to the particle ensemble.

The IBIS algorithm is closely related to the particle filter. It is initialized by drawing an ensemble of N *independent, identically distributed* (*i.i.d.*) parameter particles $\theta^{(n)},\ n = 1, \dots, N$ from the prior $p(\theta)$:

$$\theta^{(n)} \sim p(\theta)\ \forall\ n \in \{1, \dots, N\} \quad 3\text{-}7$$

which yields a particle approximation of the prior:

$$\hat{p}(\theta) = \sum_{n=1}^{N} w_0^{(n)} \delta\left(\theta - \theta^{(n)}\right) \xrightarrow{N\to\infty} p(\theta) \qquad 3\text{-}8$$

where δ is the Dirac delta function centered on $\theta^{(n)}$, $w_0^{(n)}$ is the particle's individual retrieval weight (initially a uniform $\frac{1}{N}$ starting from i.i.d. samples), and $\hat{p}(\theta)$ denotes the particle approximation of $p(\theta)$, converging towards the true prior in the limit of an infinite number of particles. Particle filters implement the Bayesian update in two steps: first, by adjusting the previous weights $w_{t-1}^{(n)}$ according to the likelihood increments $l_t^{(n)}$:

$$l_t^{(n)} = p\left(y_t|\theta^{(n)}\right) \forall\, n \in \{1, \dots, N\} \qquad 3\text{-}9$$

$$w_t^{(n)} = w_{t-1}^{(n)} l_t^{(n)} \forall\, n \in \{1, \dots, N\}. \qquad 3\text{-}10$$

In preparation for a later step, we also carry along an estimate of the total likelihood accumulated so far (initially $L_0^{(n)} = 1$):

$$L_t^{(n)} = L_{t-1}^{(n)} l_t^{(n)} = p\left(y_{1:t}|\theta^{(n)}\right) \forall\, n \in \{1, \dots, N\} \qquad 3\text{-}11$$

Second, since the updated weights of 3-10 no longer sum to unity, they must be re-normalized (3-12) to obtain the updated particle approximation of the posterior (3-13):

$$W_t^{(n)} = \frac{w_t^{(n)}}{\sum_{m=1}^{N} w_t^{(m)}} \forall\, n \in \{1, \dots, N\} \qquad 3\text{-}12$$

$$\hat{p}(\theta|y_{1:t}) = \sum_{n=1}^{N} W_t^{(n)} \delta\left(\theta - \theta^{(n)}\right) \xrightarrow{N\to\infty} p(\theta|y_{1:t}) \qquad 3\text{-}13$$

Theoretically, the steps outlined in 3-9 to 3-13 may be repeated indefinitely. Eventually, however, sample degeneracy will cause only one single particle to retain any significant weight. This issue is generally addressed through a *resampling* step, making use of the surrogate

property in 3-13 to randomly draw a set of new, equally-weighted particles. From a practical perspective, resampling duplicates well-performing particles while discarding poorly performing ones.

The original IBIS algorithm only resamples once a certain degeneracy criterion is fulfilled. With a quasi-online implementation in mind, however, we want to control the computation time and trigger a resampling and rejuvenation step after every assimilation. Resampling can be implemented in several ways, each with their own advantages and drawbacks (Li et al., 2015). In this study we employ *stochastic universal resampling* (SUR: Baker, 1987; Townsend, 2003), a method resilient to random loss of diversity during resampling (Figure 3-1). Its output is a set of indices $\left(a^{(1)}, \dots, a^{(N)}\right)$ defining which particle values each particle slots $(1, \dots, N)$ inherits (3-1):

$$a^{(1)}, \dots, a^{(N)} = \mathrm{SUR}\left(W_t^{(1)}, \dots, W_t^{(N)}\right) \in \{1, \dots, N\} \qquad 3\text{-}14$$

Mind that the number of resampled particles does not necessarily have to be equal to the previous ensemble size. In fact, we will make use of this property later to stifle growing computational demand by dynamically reducing the ensemble size. Meanwhile, however, the ancestral indices allow us to obtain a new, equally-weighted ensemble of parameter particles and their associated variables:

$$\theta^{(n)} \leftarrow \theta^{(a^{(n)})}, L_t^{(n)} \leftarrow L_t^{(a^{(n)})}, w_t^{(n)} \leftarrow 1 \qquad \forall\, n \in \{1, \dots, N\} \qquad 3\text{-}15$$

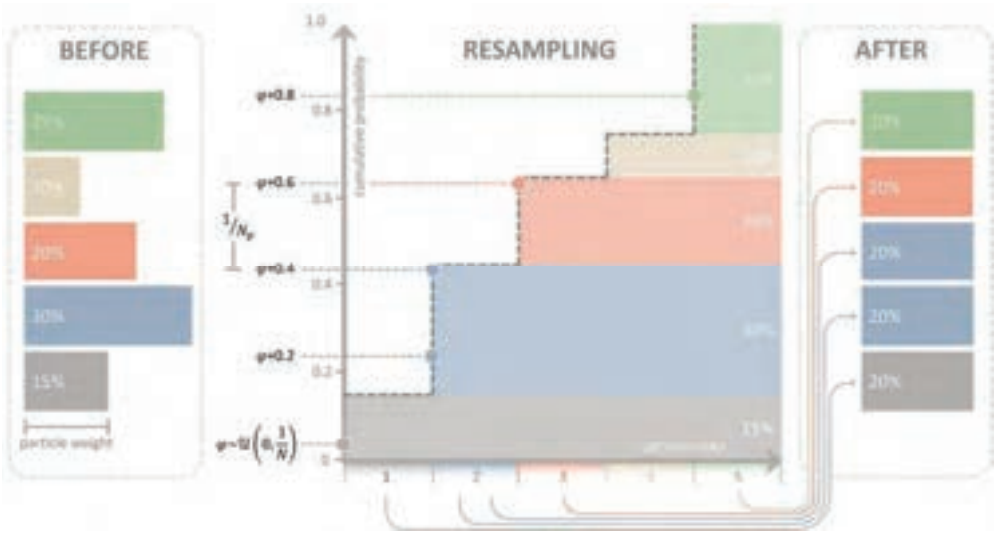


Figure 3-1. Schematic illustration of stochastic universal resampling for $N = 5$. Starting with an ensemble of non-uniformly weighted particles (left), we construct a cumulative probability function (dashed grey line, center). After drawing a random offset φ from a uniform distribution $\mathcal{U}(min = 0, max = 1/N)$, we obtain resampled particle indices by sampling this function in additive increments of $1/N$. Finally, each particle slot inherits the variables of its respective resampled particle index ($1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 2, 4 \rightarrow 3, 5 \rightarrow 5$) and the weights are reset (right, 3-15**).**

The process of resampling replaces low-weighted, unique samples with copies of high-weighted particles, reducing weight degeneracy at the cost of diversity (*sample impoverishment*: e.g., Doucet & Johansen, 2009; Li et al., 2015). While this process does not in itself solve the fundamental issue – replacing weight-based degeneracy with position-based degeneracy –, it forms a more efficient basis for *rejuvenation*, the re-introduction of diversity. Ideally, this rejuvenation mechanism should be invariant with respect to the underlying probability distribution or we risk compromising the ensemble's surrogate property. The IBIS algorithm achieves this through a *Metropolis-Hastings Markov Chain Monte Carlo* (*MH-MCMC*) jump for each particle $\theta^{(n)}$, employing a proposal density $p\left({}^{*}\theta|\theta^{(n)}\right)$:

$$ {}^{*}\theta^{(n)} \sim p\left({}^{*}\theta|\theta^{(n)}\right) \qquad \forall\, n \in \{1, \dots, N\} \qquad 3\text{-}16 $$

where the asterisk denotes the proposal, so ${}^{*}\theta^{(n)}$ is the proposal for an original particle $\theta^{(n)}$. MH-MCMC jumps are randomly accepted with a

probability defined by the ratio between the transition densities $\left(p\left(\theta^{(n)}|{}^{*}\theta^{(n)}\right)\text{ and }p\left({}^{*}\theta^{(n)}|\theta^{(n)}\right)\right)$, the prior densities $\left(p\left(\theta^{(n)}\right)\text{ and }p\left({}^{*}\theta^{(n)}\right)\right)$, as well as the total likelihoods $\left(L_t^{(n)}\text{ and }{}^{*}L_t^{(n)}\right)$, capped at one:

$$p_{accept}^{(n)} = min\left(1, \frac{p\left({}^{*}\theta^{(n)}\right){}^{*}L_t^{(n)} p\left(\theta^{(n)}|{}^{*}\theta^{(n)}\right)}{p(\theta^{(n)})L_t^{(n)} p({}^{*}\theta^{(n)}|\theta^{(n)})}\right) \forall\, n \in \{1, \dots, N\} \qquad 3\text{-}17$$

If the proposal is accepted ($v^{(n)}\sim\mathcal{U}(0,1) < p_{accept}^{(n)}$), the new particle (${}^{*}\theta^{(n)}$) and its associated variables (${}^{*}L_t^{(n)}$, ${}^{*}w_t^{(n)}$) replace the original:

$$\begin{aligned} if\ v^{(n)}\sim\mathcal{U}(0,1) < p_{accept}\text{:}\ \ \theta^{(n)} &\leftarrow {}^{*}\theta^{(n)}, L_t^{(n)} \\ &\leftarrow {}^{*}L_t^{(n)}, x_t^{(n)} \leftarrow {}^{*}x_t^{(n)} \quad \forall\, n \in \{1, \dots, N\} \end{aligned} \qquad 3\text{-}18$$

If the proposal is rejected, the algorithm continues to the next time step using the original particle and its associated variables. The evaluation of $p_{accept}^{(n)}$ is also why we carried along the total likelihood $L_t^{(n)}$ (3-11, 3-15). The most computationally expensive term in 3-17 to evaluate is ${}^{*}L_t^{(n)}$, which requires re-simulating the entire observation history up to timestep t. The cost of this evaluation increases as the assimilated time series grows, thus precluding a true online implementation. Moradkhani et al. (2012) propose a workaround by evaluating only the latest likelihood increment ${}^{*}l_t^{(n)}$ and estimating $p\left({}^{*}\theta^{(n)}|y_{1:t-1}\right)$ by evaluating a Gaussian distribution fitted to the original particles. This approach has the advantage of keeping the computational demand constant but becomes problematic if the ensemble collapses.

In this study, we instead opt to re-simulate the full observation history to obtain ${}^{*}L_t^{(n)}$ and compensate for the growing computational demand by resampling a reduced number of particles if the simulation time exceeds a user-specified threshold. Since this trick will not work

indefinitely – only until the re-simulation takes longer than the time between assimilation intervals – our IBIS-based optimizer is only quasi-online. After rejuvenation, the algorithm resumes from 3-9. With the theoretical foundations laid out, we can proceed to the algorithm's implementation.

3.4 Data and Implementation

3.4.1 Study area

Figure 3-2 illustrates various features of Settolo, the studied field site. The site is located on the eastern bank of the river Piave, near the city of Valdobbiadene in the province of Treviso, Northern Italy. Its unconfined aquifer is recharged by the nearby river and exploited for drinking water production (Zovi, 2014; Zovi et al., 2017). The surface elevation of the model domain slopes from 165 m a.s.l. in the north-west with an aquifer depth of about 30 m to about 155 m a.s.l. with an aquifer depth of about 50 m in the south-east. Assimilated water table measurements are available from 18 observation wells and two production wells from December 1st, 2010 to January 31st, 2012. The main assimilation period is from February 1st, 2011, to January 31st, 2012, the remaining data is used for validation. Of these 18 observation wells, four (*Piave Up*, *Piave Down*, *p07*, and *pSUD*) serve to inform five time-variable head boundaries (Figure 3-2b). Boundaries AB and BC are linearly interpolated from *Piave Up* and *Piave Down*, and *Piave Down* and *pSUD*, respectively, whereas A, C, and D are uniformly assigned the heads of their neighboring wells.

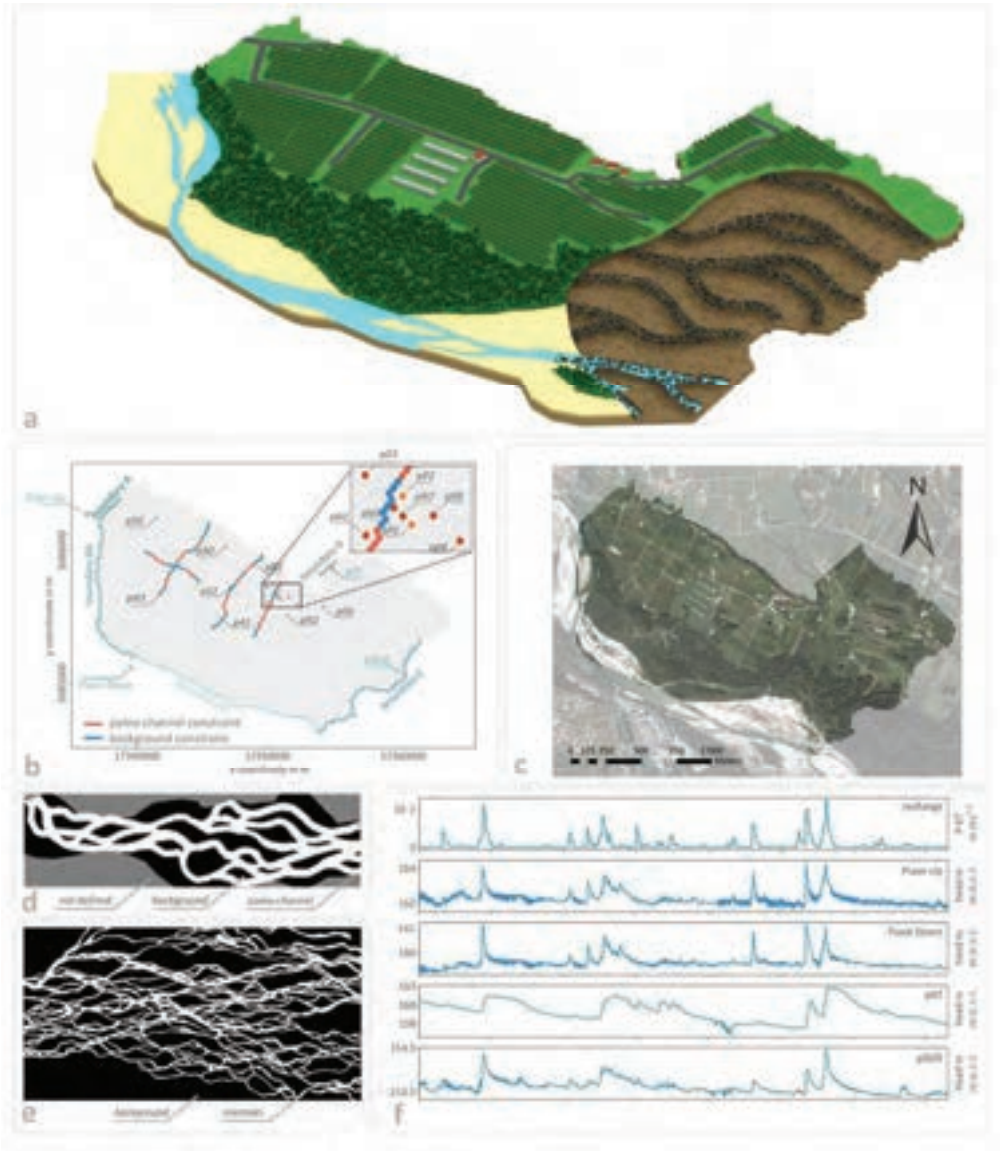


Figure 3-2. Overview of the field site in Settolo, Italy: (a) schematic render of the geological concept (not to scale); (b) grid-based map with geophysical constraints (orange and blue hexagons), observation (crimson hexagons) and pumping (yellow hexagons) wells; grey circles mark the positions of the observation wells informing the boundaries; (c) model domain overlay on a satellite image (Google Earth, 2009); (d) site-based training image for MPS simulations; (e) training image based on Skeidararsandur river, Iceland, for MPS simulations; (f) P-ET and hydraulic head in the four boundary wells over the data assimilation period.

All other boundaries are assumed to be no-flow. An irrigation water pipeline crosses the Piave riverbed between *Piave Down* and *pSUD*, acting as a physical obstacle which may cause a river level discontinuity between 0.5 m and 1 m, depending on discharge. This obstacle was not represented explicitly in the model and partially motivates the forcing model introduced in the next section. Figure 3-2f illustrates the raw forcing data over the assimilation period. Raw recharge estimates were

obtained by taking the difference between precipitation and evapotranspiration estimates after the Penman-Monteith equation using meteorological data from a nearby station in Valdobbiadene.

The (hydro)geological characterization of the site assumes the presence of highly conductive paleo-channels of the nearby river and is supported by electric resistivity tomography (ERT) measurements along a number of transects (Figure 2b, see also Zovi et al., 2017). To maintain consistency with this non-trivial geological prior as defined by a training image (Figure 3-2d), we use a multi-point statistics framework (*DeeSse*: Mariethoz et al., 2010, 2015) and constrain its realizations with the ERT transects. These constraints force the MPS framework to reproduce the prescribed facies characterization (Figure 3-2b). The training image (Figure 3-2d) was derived from satellite imagery based on the current river morphology. We note that the process of deriving a training image in such a manner can be contentious, as there is no guarantee that contemporary hydraulics are representative of the geomorphology at the time of deposition of the paleo-channels (Zovi et al., 2017). To gauge the impact of the training image, we also test an alternative image based on an unrelated fluvial system in Iceland (Figure 3-2e). Further details and information on the field site are available in Zovi et al. (2017), Zovi (2014) and at http://settolo.dicea.unipd.it/index.php.

3.4.2 Assembling the forward operator

In the following, we outline the components of the forward operator introduced in Section 3.3.2. This operator is composed of several auxiliary, deterministic modules applied in sequence. For the moment, we restrict ourselves to the overall structure and reserve greater detail for the following sections.

The first module is the *field generator* $G(\Theta)$. When pursuing conformance with a complex geological prior, it is rarely helpful to optimize the grid parameters θ directly. Instead, we optimize a set of *hyperparameters* Θ, which broadly serve as the forward operator's input and thus also parameterize the field generator. This module then creates parameter fields consistent with the geological prior, thereby ensuring conformance by construction:

$$G(\Theta) \rightarrow \theta \tag{3-19}$$

The second, perhaps more unconventional module is the *forcing model* $F(\Theta, U_{0:t})$. Groundwater dynamics are usually controlled by external driving forces such as recharge or time-variable flow across boundaries. In general, however, we cannot observe these forcings directly and must instead approximate them from related quantities. This approximation demands assumptions (e.g., instantaneous recharge, well in contact to aquifer, extrapolated head boundaries) which risk introducing systematic bias. To permit compensation for such effects, we propose a hyperparameterized pre-processor that transforms the raw forcing data $U_{0:t}$ into an updated form $u_{0:t}$:

$$F(\Theta, U_{0:t}) \rightarrow u_0, \dots, u_t \tag{3-20}$$

The third module is an inter- and extrapolation of the initial states x_0 through inverse distance weighting (*IDW*: Shepard, 1968) based on variable-head boundaries included in u_0 (which depend on Θ, see 3-20) as well as an initial set of head observations y_0:

$$\mathrm{IDW}(u_0, y_0) \rightarrow x_0 \tag{3-21}$$

The fourth deterministic module is the numerical model, which requires the previously generated grid parameters θ, the model forcing $u_{1:t}$, and initial conditions x_0 to predict the states $x_{1:t}$:

$$M(x_0, u_{1:t}, \theta) \rightarrow x_1, \dots, x_t \qquad 3\text{-}22$$

Finally, the fifth module simply extracts the predicted observations $y_{1:t}^{sim}$ from the simulated state trajectory $x_{1:t}$:

$$Hx_{1:t} \rightarrow y_{1:t}^{sim} \qquad 3\text{-}23$$

With the introduction of hyperparameters Θ as the variables of interest, we note that the theory outlined in Section 3.3 should be read in terms of the hyperparameters Θ, not the grid parameters θ. These variables completely define the error model, by parameterizing both its deterministic (the forward operator) and probabilistic (the additive Gaussian error) part. The algorithm's procedure during an assimilation increment is schematically illustrated in Figure 3-3. In the following, we will explore the modules in greater detail.

3.4.2.1 Field generator $\boldsymbol{G(\Theta)}$

The field generator is built around a facies distribution map, the output of the multi-point statistics framework DeeSse (Mariethoz et al., 2010, 2015; Straubhaar, 2019). This facies distribution map is a set of hyperparameters which defines whether a cell belongs to the paleo-channel or background sediment, then assigns grid parameters accordingly. Additional hyperparameters define the hydraulic properties of each facies: the mean hydraulic conductivity K, an internal conductivity heterogeneity map with range ΔK, and a specific yield S_y, which was assumed to be homogeneous, following Zovi et al. (2017). Figure 3-4 illustrates how we assemble the conductivity maps and create the full grid parameters θ from the two complementary parts. The

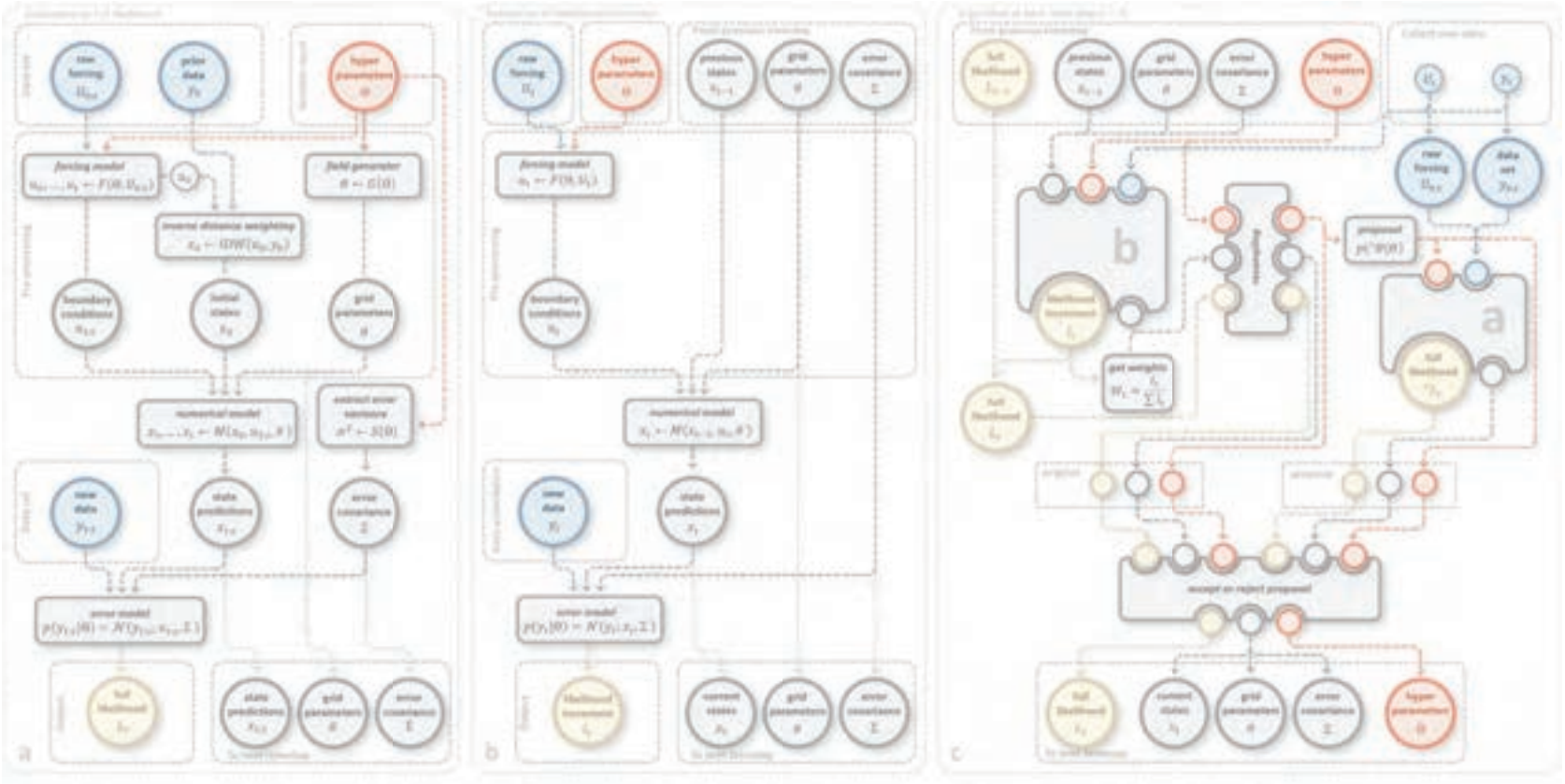


Figure 3-3. Schema for the evaluation of the full likelihood L_t (a), the likelihood increment l_t (b), and the algorithm at time $t > 0$ (c). For L_t (a), we generate the model input files x_0, $u_{1:t}$, and θ from the hyperparameters Θ and the raw forcing data $U_{0:t}$. After obtaining $x_{1:t}$, we use the assimilated data $y_{1:t}$ and the error covariance matrix $\Sigma = I\sigma^2$ to evaluate L_t. For l_t (b), x_{t-1}, θ, and Σ are already available from the previous timestep after initialization. We must only calculate u_t and can directly obtain x_t by deterministic simulation. Afterwards, we can evaluate l_t and pass x_t along as initial condition for the next timestep. The implementation of the algorithm (c) uses both a and b.

Figure 3-4. Schematic render of the field generator. We generate facies conductivity fields for the background sediment and paleo-channels by offsetting a uniform, average conductivity field with internal variability. The facies distribution map then extracts grid parameter values from each map according to the facies assignment and assembles the parameter field $\boldsymbol{\theta}$. Assignment of specific yield proceeds equivalently, barring the addition of internal heterogeneity.

heterogeneity map was generated through convolution of a white noise field with an isotropic Gaussian filter, then normalized and centred around zero, and finally scaled by ΔK.

3.4.2.2 Forcing model $\boldsymbol{F(\Theta, U_{0:t})}$

The forcing considered in this study are time-variable prescribed head boundaries (Figure 3-2b) and transient, uniform recharge (Figure 3-2f).

The pumping rates in the production wells were considered sufficiently well-quantified to warrant exclusion from the forcing model.

The forcing model is a relatively simple pre-processor illustrated in Figure 3-5. The raw forcing data $U_{0:t}$ (hydraulic heads or recharge flux) are first normalized, then transformed according to the spline defined by three hyperparameters – the spline control points –, then reverted into their canonical range (Figure 3-5a). These control points are defined independently for each boundary well and recharge.

For recharge, this process is further extended by distributing the transformed recharge $\widetilde{U}_s$ at each time s among the next λ timesteps according to an exponential distribution, where λ is an additional hyperparameter to be optimized (Figure 3-5b). Finally, the full recharge timeseries is assembled by summarizing the resulting recharge components for each timestep (Figure 3-5c). The transformed hydraulic heads and the transformed and re-distributed recharge constitute $u_{0:t}$.

The motivation for a spline transformation are: i) possible non-linearities during rainfall or high-flow events, which might cause the forcing to fit better during some meteorological regimes than during others, and ii) the possibility of bias from the inter- and extrapolation of the hydraulic heads along their respective boundaries. The transformation and redistribution of recharge is justified by large uncertainties in P and ET estimates, omission of overland flow, and vadose zone dynamics.

We note that for a real application the full time series is not available from the start. As such, U_{min} and U_{max} used during normalization could change as new extrema are recorded. This affects the transformation

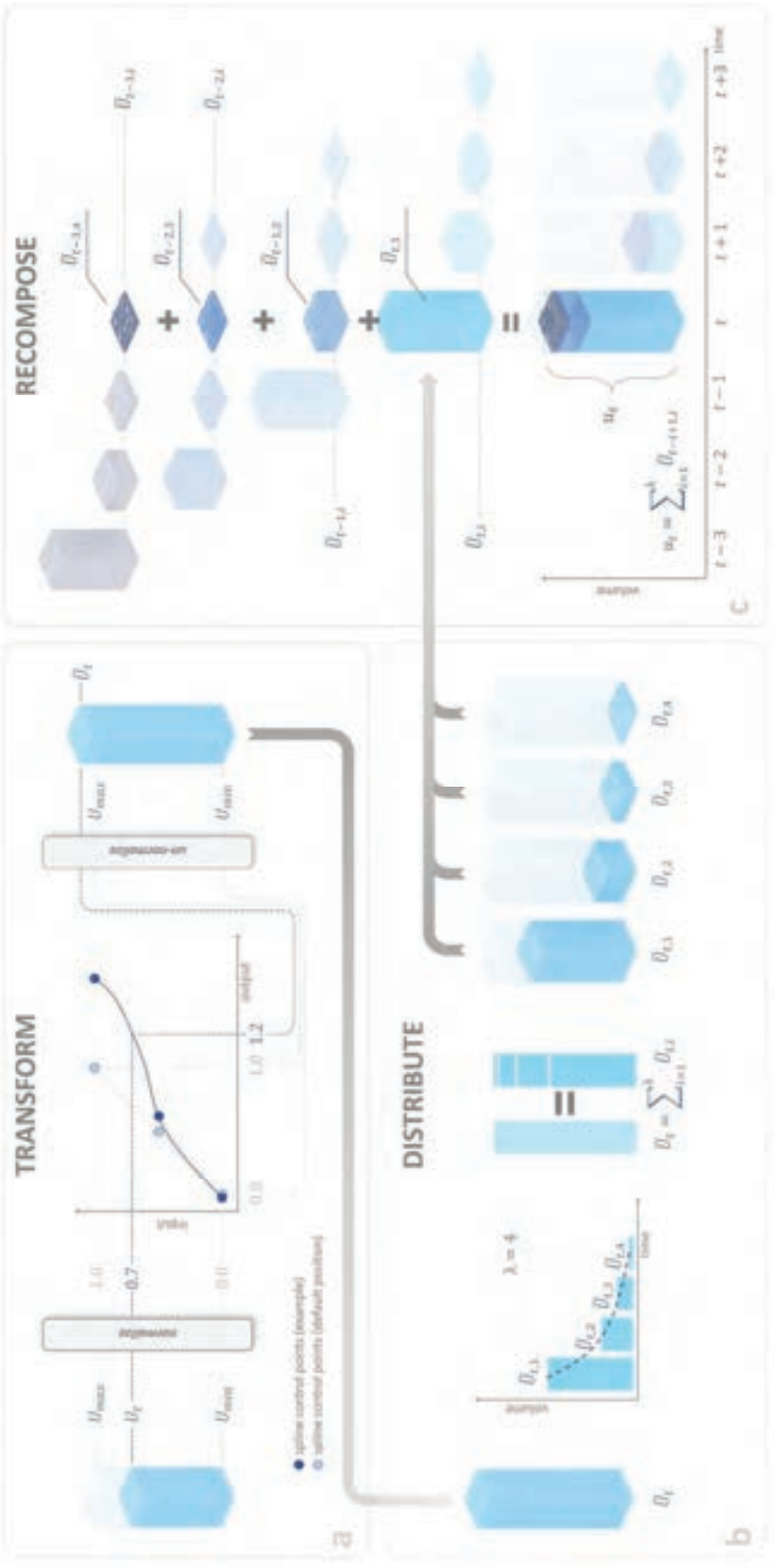


Figure 3-5. Schematic mechanism of the forcing model, with water column heights representing recharge or hydraulic head at the boundaries. The first step (a) applies to both the variable-head boundaries and the recharge: The raw quantity U_t (volume for recharge, head for boundaries) is normalized, then transformed according to a function defined by three spline control points, then reverted back to obtain the transformed forcing $\bar{U}_t$, which may lie outside the interval of U_{min} and U_{max}. For groundwater recharge, $\bar{U}_t$ is then distributed between the next λ

between canonical and normalized space and as such the value ranges affected by the spline control points, which are defined in normalized space. In real data assimilation scenarios, we thus recommend specifying a sufficiently broad U_{min} and U_{max} a-priori.

3.4.2.3 Inverse distance weighting $IDW(\boldsymbol{u_0}, \boldsymbol{y_0})$

The initial states are generated with inverse distance weighting according to Shepard (1968). This approach extrapolates from a collection of N_v known points $v_0 = \{y_0 \cup u_0\}$ to any of the N_{cell} grid points based on spatial distance $d(i,j)$ between cells i and known points j as well as a power factor p (in our case $p = 3$).

3.4.2.4 Numerical model $\boldsymbol{M(x_0, u_{1:t}, \theta)}$

The field site is simulated as a two-dimensional, unconfined, transient groundwater model using MODFLOW-USG (Panday et al., 2013) with the Python interface FloPy (Bakker et al., 2016). Its full governing equation describes water balance conservation on an infinitesimally small control volume:

$$\frac{\partial}{\partial x}\left(K_x \frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial h}{\partial z}\right) = S_s \frac{\partial h}{\partial t} + Q_s \qquad 3\text{-}24$$

where $K_{x/y/z}$ are the (possibly heterogeneous and anisotropic) hydraulic conductivities in each spatial dimension, Q_s a source-sink term, and S_s is the specific storage of the porous medium. MODFLOW solves these equations through a finite volume approach (Langevin et al., 2017). The model domain is discretized with uniform, hexagonal cells. Since DeeSse demands a regular grid, we further define a regular but anisotropic support grid which contains the hexagonal cell centers for the MPS simulation. Cells with extraction wells are further subdivided to increase grid resolution around the cones of depression (Figure II-1). The subdivided cells are not resolved for the purpose of optimization and

inherit their host cell's parameterization. The cell count without the subdivision is 12,856.

3.4.3 Probabilistic setup

3.4.3.1 Hyperparameter priors

Since we seek to optimize the hyperparameters Θ, we also define the model prior in terms of these variables. Summing over Section 3.4.2, we have a total of $3N_{cells} + 23$ hyperparameters – three hyperparameter fields (the facies distribution, and two independent internal variability maps), and 23 scalar hyperparameters. The hyperparameter limits and priors are listed in Table 3-1.

3.4.3.2 Rejuvenation mechanism

As described in Section 3.3.3, we rejuvenate the ensemble with MH-MCMC jumps. This procedure requires a proposal distribution $p\left({}^{*}\Theta|\Theta^{(n)}\right)$ to suggest new values ${}^{*}\Theta$ for an original particle $\Theta^{(n)}$. While we could update all hyperparameters at once, doing so is not always useful – to achieve a good acceptance rate, it is important to keep the magnitude of the mutation sufficiently low. We achieve this by defining $p\left({}^{*}\Theta|\Theta^{(n)}\right)$ as a mixture distribution of random multivariate normal mutations which only update a subset of hyperparameters at a time. Independently for each particle, we either update the hydraulic hyperparameters ($K, \Delta K, S_y$: 65%), the spline control points for recharge (SCP_{rech}: 7.5%) or one of the boundary wells (SCP_{well}: $3.75\% * 4 = 15\%$), the recharge delay λ (7.5%), or the error standard deviation σ (5%). The proposal standard deviations and correlation structures between hyperparameters are listed in Table 3-1.

Furthermore, we must account for possible asymmetries in the proposal $\left(\text{i.e., } p\left({}^{*}\Theta^{(n)}|\Theta^{(n)}\right) \neq p\left(\Theta^{(n)}|{}^{*}\Theta^{(n)}\right)\right)$, which arise for example in the

Table 3-1. Hyperparameter limits, priors, and proposal distributions for the scalar hyperparameters. Limits are assigned only to the non-negative recharge delay λ and model error σ, as well as the hydrogeological hyperparameters $K, \Delta K$ and S_y, where subscript '1' denotes the paleo-channel and subscript '2' the background sediment. The priors are either uniform (with the assigned limits), Gaussian (with mean in brackets, and listed standard deviation), or exponential (with listed factor α). For the proposal distribution, some hyperparameters are updated jointly with a multivariate Gaussian of listed standard deviation and correlation structure. The limits for the hydraulic parameters were adopted from Zovi et al. (2017).

Hyperparameter [units]	limits		prior pdf			proposal distribution	
	min	max	uniform [see limits]	Gaussian [μ / σ]	exponential [α]	standard deviation	correlation matrix
K_1 [$\log_{10}$ m/s]	-2.3	-1.9	yes			0.05	
K_2 [$\log_{10}$ m/s]	-3.3	-2.7	yes				
ΔK_1 [$\log_{10}$ m/s]	0.1	0.2	yes			0.005	
ΔK_2 [$\log_{10}$ m/s]	0.2	0.5	yes				
$S_{y,1}$ [−]	0.1	0.35	yes			0.025	
$S_{y,2}$ [−]	0.1	0.35	yes				
$SCP_{weld}(0.0)$ [−]				0.0 / 0.01		0.001	
$SCP_{weld}(0.5)$ [−]				0.5 / 0.01		0.05	
$SCP_{weld}(1.0)$ [−]				1.0 / 0.01		0.1	
$SCP_{rech}(0.0)$ [−]				0.0 / 0.001		0.001	
$SCP_{rech}(0.5)$ [−]				0.5 / 0.05		0.05	
$SCP_{rech}(1.0)$ [−]				1.0 / 0.1		0.1	
λ [3h]					1	1	
σ [m]					0.025	0.01	

presence of hyperparameter limits. Quantifying these asymmetries would require evaluating truncated, correlated, multivariate Gaussians pdfs, which is not a trivial task. Instead, we simplify this issue by interpreting hyperparameter limits as reflective boundaries, then summing the probability densities of all different jumps which would result in equivalent proposals (see Figure II-2). Since this would require evaluating probability densities of an infinite series of mirrored positions along dimensions with both a lower and an upper limit, we restrict our evaluation only to the probability densities of the first reflection across each parameter limit. For practical application, we reflect any potential proposal falling outside the hyperparameter limits across its closest limit until it falls within the hyperparameter constraints.

The hyperparameter maps (facies distribution, internal variability) are updated together with the hydraulic hyperparameters. The facies distribution is updated with a blocking-moving window procedure

similar to Alcolea & Renard (2010): first, we select a random cell, then remove all entries in a random radius between 75 m and 500 m around this cell. After this, the facies distribution within the masked circle is re-generated with MPS, with the remaining facies assignments outside acting as constraints to ensure continuity of the features. The internal variability maps are updated through small, gradual changes: first, each map is normalized and re-centered around zero. Then a new variability map is generated (see Section 3.4.2.1), scaled by 0.25, and added to the previous map. The result is then renormalized, re-centered, and scaled with its respective (updated) ΔK.

3.4.4 Scenarios and computational setup

In this study we consider three different scenarios: Scenario 1 uses the training image based on the local field site (Figure 3-2d) and the forcing model. Scenario 2 is the same as Scenario 1 but replaces the field-based training image with the alternative training image (Figure 3-2e). Scenario 3 differs from Scenario 1 only in omission of the forcing model. We initialized each scenario with 350 particles, then allowed the algorithm to dynamically adjust the ensemble size to meet the prescribed computational limits.

The optimization algorithm is implemented in Python 3.3.2 in a parallelized framework with eight workers on three workstations. Simulations for each scenario are repeated in triplicate with different random seeds to test reproducibility: the scenarios with suffix 'a' were simulated on a Lenovo ThinkPad X1 Carbon with an Intel® Core™ i7-6600 CPU with two cores (four logical processors) @2.60 GHz and 16 GB of RAM, the simulations with suffix 'b' on a workstation with an Intel® Core™ i7-3770 CPU with four cores (eight logical processors) and 8 GB

of RAM, and the simulations with suffix 'c' on a workstation with an Intel® Core™ i7-2600 CPU with four cores (eight logical processors) @3.40 GHz and 8 GB of RAM.

The MPS framework we used was DeeSse (Mariethoz et al., 2010, 2015; Straubhaar, 2019), a commercial direct sampling MPS software freely available for scientific use. The communication between DeeSse and our optimizer was established with a self-designed Python interface. Data was assimilated every 3 h, with resampling and rejuvenation steps being triggered every 24 h of data. The allocated simulation time for 24 h worth of data was bounded between 600 and 900 s – considering 365 days of data, on average a little over three days –, well below the time available in a real application. Before proceeding to the presentation of the results, we will introduce the performance metrics used.

3.4.5 Performance metrics

The *root-mean-square error* (*RMSE*) is calculated for each observation well $o = 1, \dots, N_{obs}$ according to

$$RMSE^o = \sqrt{\frac{1}{N}\frac{1}{t}\sum_{n=1}^{N}\sum_{s=1}^{t}\left(y_s^{sim,o,(n)} - y_s^o\right)^2} \qquad 3\text{-}25$$

and subsequently averaged across all observation wells

$$\overline{RMSE} = \frac{1}{N_{obs}}\sum_{o=1}^{N_{obs}} RMSE^o \qquad 3\text{-}26$$

while the *bias* is calculated according to

$$\overline{bias} = \frac{1}{N}\frac{1}{N_{obs}}\frac{1}{t}\sum_{n=1}^{N}\sum_{o=1}^{N_{obs}}\sum_{s=1}^{t}\left(y_s^{sim,o,(n)} - y_s^o\right) \qquad 3\text{-}27$$

The *percentage bias* and the *Kling-Gupta efficiency* (*KGE*: Gupta et al., 2009) are two performance metrics popular in surface hydrology. Both metrics require a reference level, which is not clearly defined for groundwater tables. In this study, we use the aquifer bottom at each observation well z^o as the reference level:

$$\overline{pbias} = \frac{1}{N}\frac{1}{N_{obs}}\frac{1}{t}\sum_{n=1}^{N}\sum_{o=1}^{N_{obs}}\sum_{s=1}^{t}\left(\frac{y_s^{sim,o,(n)} - y_s^o}{y_s^o - z^o}\right) * 100 \qquad 3\text{-}28$$

$$\overline{KGE} = \frac{1}{N_o}\sum_{o=1}^{N_o}\left(1 - \sqrt{(r^o - 1)^2 - (\alpha^o - 1)^2 - (\beta^o - 1)^2}\right) \qquad 3\text{-}29$$

where r^o is the Pearson correlation coefficient, α^o the ratio of standard deviations and β^o the ratio of means (against the reference level) between the simulated and observed time series at observation well o. The mean Spearman correlation coefficient (Spearman, 1987) was averaged across all observation wells:

$$\overline{r_{sp}} = \frac{1}{N_o}\sum_{o=1}^{N_o} r_{sp}^o \qquad 3\text{-}30$$

where r_{sp}^o was obtained with the python library scipy.stats.

3.5 Results

The simulation results are evaluated at the end of the data assimilation period (day 365) using the final hyperparameter ensemble and the performance metrics described in section 3.5. We report these metrics over the 365-day assimilation period (from February 1st, 2011 to January 31st, 2012) and the 90-day validation period (from December 1st, 2010 to February 28th, 2011) used in Zovi et al. (2017). Since this validation period overlaps for the last month with the assimilation period – a decision

made since the validation set's observation time series otherwise would not extend to all wells –, we also evaluated a shortened, non-overlapping validation period spanning only 60 days (from December 1st, 2010 to January 29th, 2011) for reference. The resulting metrics for all three scenarios and evaluation periods are depicted in Figure 3-6 and listed in Table S2.

3.5.1 Performance metrics

In general, most scenarios display relatively similar performance metrics (Figure 3-6, Table S2). The overall model fit is very satisfying across all scenarios (Figure 3-7, Figures S3 to S11). On average, Scenario 1 yields the best $\overline{RMSE}$ over the assimilation (0.131 m), validation (0.136 m), and shortened validation (0.138 m) periods, closely followed by Scenario 2 (0.134 m, 0.151 m, and 0.139 m). Scenario 3 performs slightly worse (0.140 m, 0.158 m, and 0.164 m) than the other two scenarios. The absolute values of $\overline{bias}$ and $\overline{pbias}$ are very low across the assimilation period in all scenarios. Over the validation periods, biases are slightly larger, likely owed to the increased prominence of the rain event and its subsequent recession (Figure II-3). Mean $\overline{KGE}$s over the assimilation period are also favourable, ranging from 0.864 (Scenario 3) to 0.916 (Scenario 1), but are somewhat lower over the validation (0.767 in Scenario 3 to 0.845 in Scenario 1) and shortened validation (0.781 in Scenario 3 and 0.877 in Scenario 1) periods. We note that $\overline{KGE}$s are lower in Scenario 3, owed primarily to a lower α value in this scenario (Tables S2 to S4). The mean Spearman correlation coefficients $\overline{r_{sp}}$ are generally very high (> 0.90), indicating strong control by the boundaries, and, curiously, are somewhat higher in the validation periods than in the assimilation period.

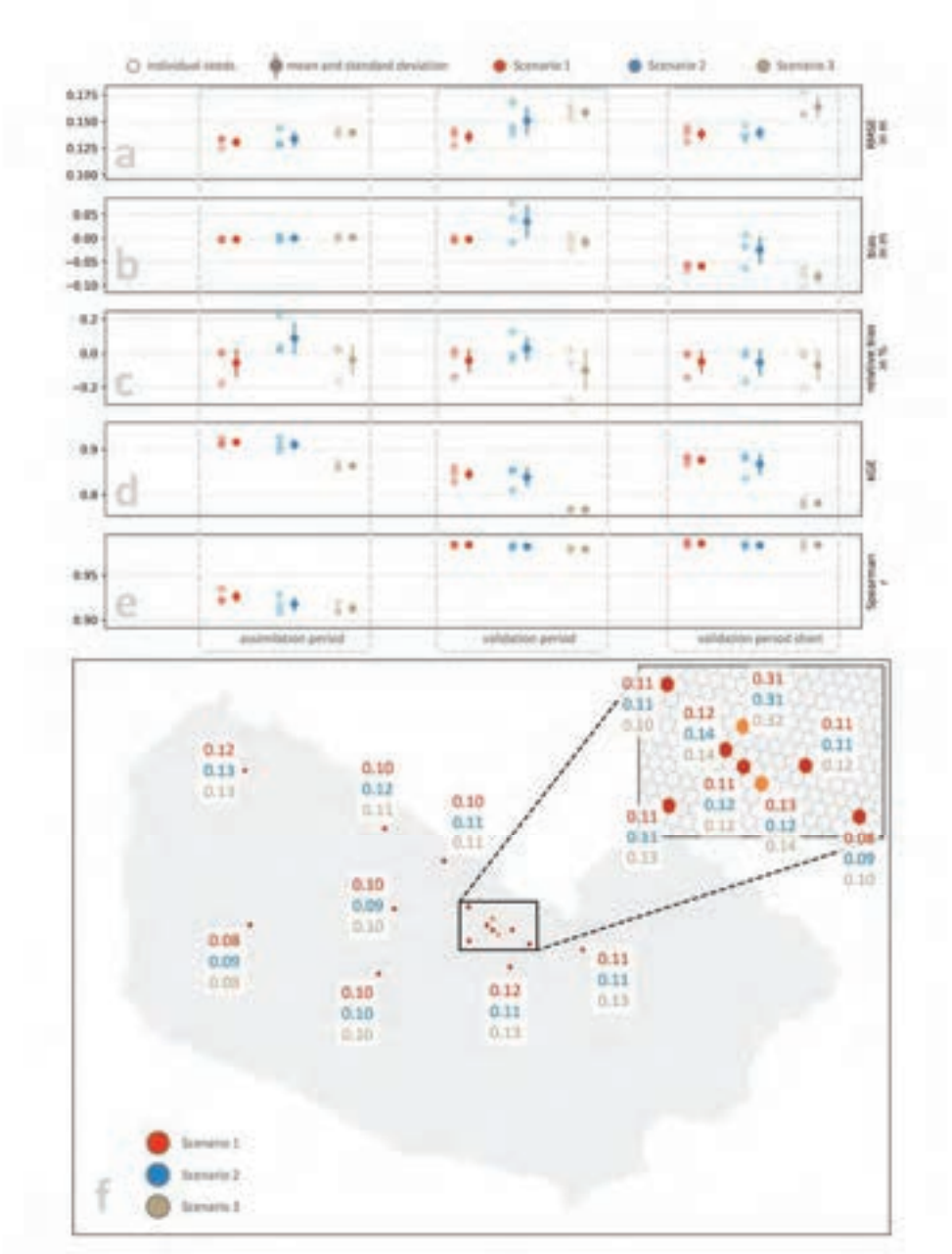


Figure 3-6. Performance metrics for the posterior ensemble of hyperparameters at the end of the 1-year assimilation period, evaluated over the assimilation period, the 90-day validation period, and the shortened 60-day validation period. Individual seeds, mean, and standard deviation are plotted for RMSE (a), bias (b), relative bias (c), KGE (d) and Spearman r (e). The lowermost subplot (f) depicts average RMSEs over the assimilation period for all scenarios and observation (red) and pumping (orange) wells. Further detail is provided in tabular form in Table S1.

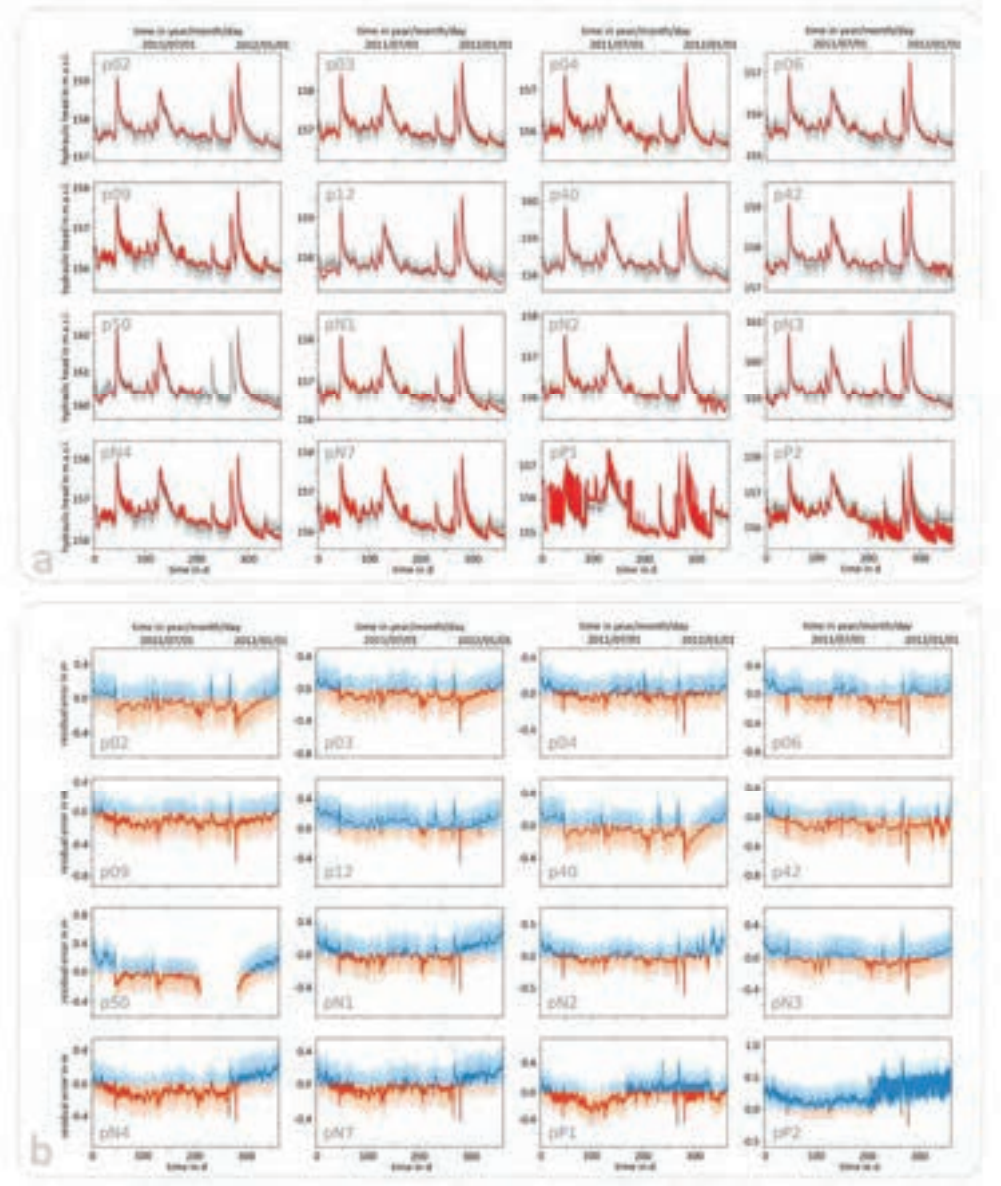


Figure 3-7. Evaluation of the model predictions of Scenario 1c over the assimilation period for all observation and pumping wells. (a) illustrates the observations (red line) and simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation. (b) illustrates the corresponding residual errors, colored according to positive (blue) and negative (red) residual components. Equivalent figures for the other scenarios and periods, as well as videos illustrating the evolution of the model prediction during assimilation are available in Figure II-3 to Figure II-11.

Analyzing the well-specific $RMSE^o$s over the assimilation period reveals a significant error in the second pumping well (*pP2*, Figure 3-6f, northern pumping well). A look at Figure 3-7b reveals that the hydraulic head in this well is systematically over-predicted and indicates that this error is exacerbated during periods of intense pumping. This could suggest that in this well the connection of the water table to the surrounding aquifer

is weakened, possibly due to a local region of low conductivity, a clogged well screen, or skin effects during pumping (e.g., Barrash et al., 2006). It is furthermore possible that the vicinity of the geophysical constraints (see Figure 3-2b) might not have left the MPS algorithm sufficient flexibility to possibly generate a more promising facies constellation.

Finally, we observe that the error residuals seem to be significantly auto-correlated across time and space (Figure 3-7b), with a pronounced trough – indicating underprediction – from roughly April to November, or (conversely) an overprediction from November to April. Similar start- and endpoints over the assimilation period suggest this could be due to seasonal phenomena not accounted for.

3.5.2 Scalar hyperparameters

Figure 3-8 illustrates the hyperparameters (and their uncertainty, when applicable) for all scenarios along the optimization process. The first and third subplot (Figure 3-8a and Figure 3-8c) visualize the ensemble size adjustments to maintain a bounded computational effort. The effective ensemble size in terms of unique i.i.d. samples (Figure 3-8b, semi-transparent) is generally about an order of magnitude lower than the raw ensemble size (Figure 3-8b, full). The second subplot (Figure 3-8b) illustrates the time-averaged acceptance chance for rejuvenation proposals, which is markedly lower for Scenario 3 – the scenario without access to the forcing model. The error standard deviation (Figure 3-8d) reveals a similar trend across all scenarios and random seeds, matching the similar performance metrics obtained in Section 3.5.1, yielding final values between 0.13 and 0.15 m. The average hydraulic conductivity of the paleo-channel facies (Figure 3-8e) is strongly constrained by its limits, but Scenarios 1 and 3 converge, on average, towards a slightly lower

value than Scenario 2. This might be rooted in the lower channel-to-background ratio of the alternative MPS training image in Scenario 2. The average conductivity of the background (Figure 3-8f) is similar across all scenarios, converging against the upper bound. The internal variability ranges of the paleo-channel (Figure 3-8g) and background facies (Figure 3-8h) show no significant trend across the different scenarios. Specific yield (Figure 3-8i) for both facies quickly converges towards the lower bound across all scenarios, likely to achieve the simulated drawdown. The forcing-related hyperparameters (Figure 3-8j to Figure 3-8o) are only used in Scenarios 1 and 2. The parameter for recharge delay seems to be unnecessary, as it converges towards the lower bound of 1 (i.e., no delay) for both scenarios. The recharge spline control points (Figure 3-8k) seemingly feature a common optimum: the upper control point converges to values around 1.65 for both scenarios, whereas the central control point drops to a value of 0.35. Physically interpreted, these values increase the flashiness of the recharge. For the boundary wells, the spline control points display no clear, significant optimum but seem to faintly increase the hydraulic head in the river-based wells. Combining this finding with the increased proposal acceptance rate in Scenarios 1 and 2 (Figure 3-8b), and the overall greater degree of uncertainty of recharge-related parameters (Figure 3-8j, k) suggests that updates to the recharge control parameters have only moderate effect on the predictions. Updates to the hydraulic parameters, the model error, or the boundary well control points seem to induce larger changes, resulting in lower acceptance rates

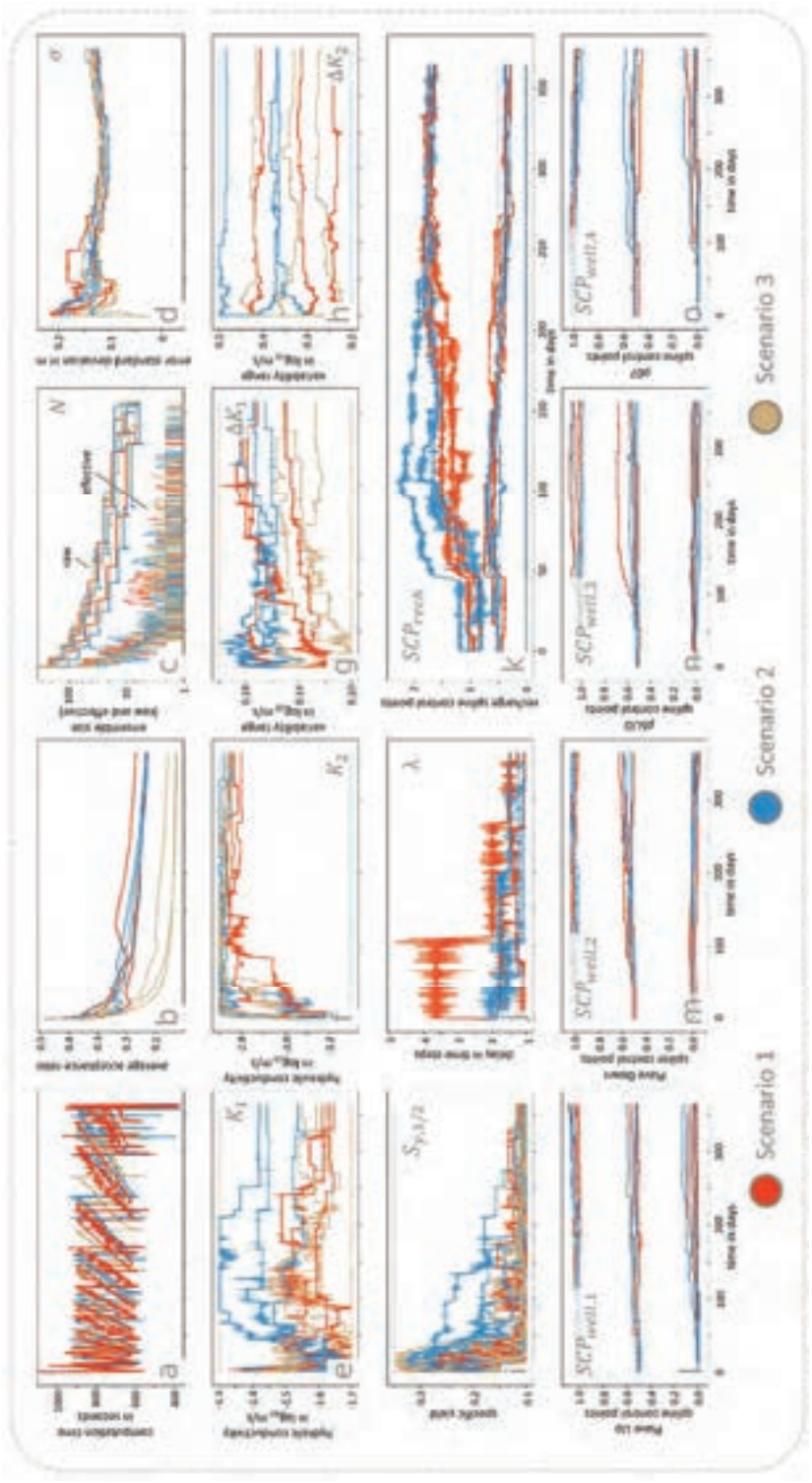


Figure 3-8. Meta- and hyperparameters for all scenarios over time: computation time per day (a), time-averaged proposal acceptance ratio (b), raw and effective particle count (c), error standard deviation (d), mean hydraulic conductivity for the paleo-channels (e) and the background sediment (f), their respective internal variability ranges (g and h), specific yield for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable. Individual corresponding hyperparameter plots are available in the Figure II-12 to Figure II-20. Subplots (a) through (c) are reproduced in greater detail in Supporting Information S21

3.5.3 Hyperparameter fields

Since conformance to a prescribed geology was a major objective of this study, investigating the hydraulic conductivity and facies maps at the end of the assimilation period (Figure 3-9) can reveal interesting insights into the optimization process.

First off, the different characteristics of the two training images become clear already when investigating the prior conductivity maps (Figure 3-9a, d, g): the smaller, local training image (Scenarios 1 and 3) generates fewer distinct variations of its patterns and thus displays clear preferences for prior channel placement, whereas the larger, alternative training image (Scenario 2) contains a wider range of possible channel constellations, resulting in much less preferential initial facies assignments.

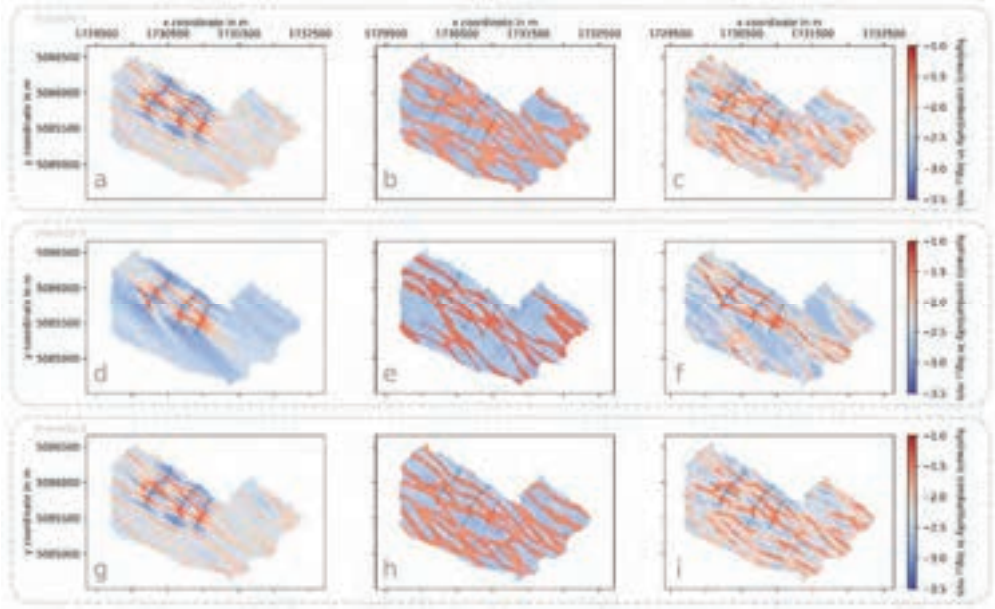


Figure 3-9. Representative ensemble mean parameter fields of Scenarios 1 (a, b, c), 2 (d, e, f), and 3 (g, h, i) at the start (a, d, g) and end (b, e, h) of the data assimilation period, for the third random seed (suffix c) each. The right-most column (c, f, i) depicts the mean conductivity field across all three random seeds for each scenario.

Investigating the expected conductivity fields at the end of the assimilation period (Figure 3-9b, e, h) reveals that the realizations retained connected features through the optimization process. We further note differences in the mean hydraulic conductivities: as suspected in Section 3.5.2, Scenario 2 indeed identifies higher facies conductivities to compensate for its lower channel-to-background facies ratio.

The final column (Figure 3-9c, f, i) serves to reveal any potential common facies assignments across the different random seeds, scenarios, and training images. Evidently, it seems that there is no clear preference aside from the prescribed ERT constraints. This can partially be explained in the relatively low information content water tables alone provide for the placement of preferential flow paths.

However, we noted that changes to the facies distribution map have significant impact on the magnitude of the likelihood ratios (Figure II-22 to Figure II-24), suggesting that the facies distribution is far from insensitive. Combining this observation with the lack of structural uncertainty in each individual scenario, and the different constellations identified across all scenarios, suggests the presence of multiple, isolated optima at least in the facies distribution hyperparameter-subspace.

3.6 Discussion

Summarizing the results, we found that in all scenarios considered the proposed quasi-online optimizer proves capable of identifying promising hyperparameter sets, while honouring a complex geological prior as prescribed by a training image. Simulations in triplicate suggest that the algorithm reliably identifies global optima for most of the scalar hyperparameters, barring the tightly-constrained internal variability

ranges ΔKs. The boundary spline control points deviate slightly but do not display a clear preference towards a global optimum either. This might be owed to interactions with the facies distribution map, which seems characterized by multiple, isolated optima.

Despite restricting the optimization to the support of the geological prior, we achieved results similar or even favourable to Zovi et al. (2017). Over the validation period, they obtained $\overline{RMSE}$s of 0.155 m without and 0.302 m with normal-score transform using the ensemble mean. However, their best-performing scenario could maintain only limited conformance to the geological structure, with no clear identification of the paleo-channels. Our most similar setup – Scenario 3, without the forcing model – yielded $\overline{RMSE}$ s between 0.153 and 0.165 m, with a plausible paleo-channel pattern that honors the available geophysical data. We found that further improvements could be made by introducing a (hyper)parameterized forcing model, which reduced $\overline{RMSE}$s down to values between 0.128 m and 0.142 m (Scenario 1). Its introduction permitted an increase of recharge intensity during precipitation events (Figure 3-8j) which brought simulated heads closer to the observations (Table S3). We note, however, that we could not use the exact same prior as Zovi et al. (2017), and acknowledge that their EnKF implementation was more computationally efficient (requiring only 30 wall-clock hours on a four-core machine) than our algorithm.

Combining the forcing model with a different training image (Scenario 2) slightly deteriorates the performance metrics relative to Scenario 1, yielding $\overline{RMSE}$s between 0.139 m and 0.168 m. An analysis of the residuals (Figure 3-7, Figure II-3 to Figure II-11) suggests that a more flexible forcing model – capable of transforming not only the raw input data but adding seasonal effects – might be necessary to attain further

improvements. A possible explanation for this apparent seasonality might be recharge from irrigation, a practice known to be used in this region (D'Agata, 2019; Zovi, 2014) but for which no data are available. D'Agata (2019) mentions compacted sands with 'low water reserves' as the cause for the need for irrigation, which might match the low S_y identified across all scenarios. Further simulation inadequacies are the large $RMSE^o$s in pumping well pP2, which might be addressed by permitting the model to consider well skin effects.

A limitation of this study is ensemble collapse. Across all scenarios, the effective ensemble size is about one order of magnitude smaller than the raw ensemble size (Figure 3-8c). This results in an underestimation of parameter uncertainty, the potential loss of separate posterior modes, and exacerbates the risk of entrapment in a local optimum. A quick back-of-the-envelope calculation reveals that this is a consequence of highly-tapered posterior modes, in turn a result of our error model definition:

Assume two hyperparameter sets creating predictions of different fidelity: the first one's predictions are always $0.01\ m$ off the observations in each well, the other's always $0.02\ m$. Further assume that $\sigma = 0.15\ m$ and that we assimilate data in 16 wells every 3 h for a year ($365 \cdot 8 \cdot 16 = 46720\ observations$). The log-likelihood difference between the two parameter sets would be -311.5. This effect becomes more pronounced for greater deviations from the observations.

This means that even for small differences in performance the rejuvenation mechanism accepts only strict improvements. Combining this finding with the presence of isolated optima indicated by the results across different random seeds suggests that the posterior might be both

highly tapered and multimodal, which renders the identification of multiple modes at once with a particle filter highly challenging.

We argue that ensemble collapse – while clearly undesirable – has less irreversible repercussions for the IBIS algorithm than for classic particle filters. Without relying on the fidelity of the ensemble approximation, the rejuvenation mechanism samples the posterior directly. This critical property permits the algorithm to recover samples from the true posterior even after ensemble collapse. As possible remedies, the steep divergence of the likelihoods (see Footnote 1) could be slowed through adjustments to the error model, such as the consideration of temporal and spatial correlations or the use of a likelihood function with heavier tails (e.g., a Cauchy distribution: van Leeuwen, 2003). From an algorithmic perspective, adjustments to the MH-MCMC proposal – e.g., the introduction of elements from evolutionary algorithms (e.g., Abbaszadeh et al., 2018; Zhu et al., 2018) – might permit the proposal to capitalize on information of other ensemble members to better identify multiple optima. We could also adapt the proposal distribution dynamically during assimilation in order to adjust acceptance rates. Such efforts, however, would be complicated by the conflicting findings of Section 3.5.2 and 3.5.3: particularly for the hydraulic parameters, larger proposals may be required to escape local minima, and smaller proposals may be required to increase the proposal acceptance rate. This contradiction is a well-known challenge in MCMC literature (e.g., Foreman-Mackey et al., 2013; Holmes et al., 2017; Tjelmeland & Hegstad, 2001).

Alternatively, ad-hoc solutions based on artificial variance inflation (e.g., Moradkhani et al., 2005) remain the most efficient remedy for the negative repercussions of ensemble collapse. Such methods might

prevent or even reverse the uncontrolled tapering of probability densities. Unfortunately, despite their pragmatic allure, such approaches invariably corrupt the posterior (e.g., Vrugt et al., 2013) and should thus be used with caution.

However, it may be an interesting direction for future research to explore the interaction of variance inflation through artificial random parameter dynamics (e.g., Moradkhani et al., 2005; Ramgraber et al., 2019) with the rejuvenation mechanism used in this study. While the indiscriminate addition of random components will corrupt the posterior, we expect that the inclusion of MCMC steps might limit posterior drift.

We believe that results of this study demonstrate the ability of the IBIS algorithm to optimize complex hydrogeological models under field conditions, although the loss of parameter uncertainty remains a major concern. The inclusion of a forcing model furthermore extended the potential for bias correction from the grid parameters to boundary conditions. This improved results slightly and permitted a more detailed diagnosis of residual model inadequacies. Considering the prevailing uncertainties in meteorological and hydrological forcing, we advocate the careful use of such pre-processors as a valuable extension to the conventional scope of hydrogeological parameter inference, if designed with physical processes or error-diagnostic capabilities in mind.

3.7 Conclusions

Reconciling the challenges posed by complex geological priors with the operational limitations of online optimization frameworks is not a trivial task, but a necessary endeavor if such methods should ever find use outside of simple settings. In pursuit of such a framework, we presented a quasi-online optimizer based on the IBIS algorithm. Instead of

optimizing grid parameters directly, we updated a set of hyperparameters which parameterized a number of pre-processors. We used a field generator built around MPS facies maps and the flexibility of the MH-MCMC rejuvenation mechanism to maintain conformance with a non-trivial geological prior by construction. We further introduced another pre-processor to reduce forcing-related biases and inadequacies. We demonstrated the performance of the algorithm with data from a field site in northern Italy.

Optimization results were promising, identifying similar hyperparameter optima across all scenarios. Despite remaining confined to the support of the geological prior, the performance metrics we obtained revealed equivalent or even superior performance to a previous study using the EnKF (Zovi et al., 2017). The use of a local training image as well as the forcing model both improved the predictions. Analysis of the error residuals further suggests forcing-related inadequacies, particularly omitted seasonal effects possibly linked to irrigation, which our pre-processor was not equipped to compensate. Allowing the forcing model to correct for unaccounted seasonality might be required to achieve further improvements.

To conclude, we believe that the ability to sequentially optimize parameter fields with non-trivial priors while providing estimates of predictive uncertainty could prove a valuable asset to practitioners in the future. Considering the larger time window available for an assimilation step in practice – 24 hours as opposed to 15 minutes – we believe there is sufficient computational space for larger ensembles or more complex models. A possible limitation of this study is the lack of structural uncertainty in the MPS facies distribution map, despite the identification of separate, functionally similar optima across different random seeds.

This suggests the presence of multiple, isolated, highly-tapered optima in the posterior pdf, which prove a highly-challenging task for most Bayesian inference algorithms. We suggested adjustments to the error model, the proposal function, or variance inflation as possible remedies. Continuing research could investigate some of these avenues in order to provide better structural uncertainty estimates. Combining the forcing model with information related to vegetation activity might furthermore permit estimates of irrigation activity, a factor which is otherwise difficult to quantify.

3.8 Acknowledgements

We express our gratitude to Dr. Julien Straubhaar, University of Neuchâtel, for providing support during the implementation of DeeSse. We furthermore thank Prof. Peter Reichert, Dr. Carlo Albert, and Andreas Scheidegger, Swiss Federal Institute of Aquatic Science and Technology (Eawag), for providing valuable advice during the conceptualization phase of this study. Finally, the first author would like to thank the Department of Civil and Environmental Engineering, University of Padova, for hosting him during a secondment. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675120.

4 Non-Gaussian parameter inference for hydrogeological models using Stein Variational Gradient Descent

4.1 Abstract

The sustainable management of groundwater demands a faithful characterization of the subsurface. This, in turn, requires information which is generally not available. To bridge the gap between data need and availability, numerical models are often used to synthesize plausible scenarios not only from direct information but also additional, indirect data. Unfortunately, the resulting system characterizations will rarely be unique. This poses a challenge for practical parameter inference: Computational limitations often force modelers to resort to methods based on questionable assumptions of Gaussianity, which do not reproduce important facets of ambiguity such as Pareto fronts or multi-modality. In search of a remedy, an alternative could be found in Stein Variational Gradient Descent, a recent development in the field of statistics. This ensemble-based method iteratively transforms a set of arbitrary particles into samples of a potentially non-Gaussian posterior, provided the latter is sufficiently smooth. A prerequisite for this method is knowledge of the Jacobian, which is usually exceptionally expensive to evaluate. To address this issue, we propose an ensemble-based, localized approximation of the Jacobian. We demonstrate the performance of the resulting algorithm in two cases: a simple, bimodal synthetic scenario, and a complex numerical model based on a pre-alpine catchment. Promising results in both cases – even when the ensemble

size is smaller than the number of parameters – suggest that Stein Variational Gradient Descent could be a valuable addition to hydrogeological parameter inference.

4.2 Introduction

Parameter estimation of numerical models can synthesize different types of information into a physically plausible narrative. This is of particular relevance for the discipline of hydrogeology, where informed management demands detailed knowledge of the system, but direct measurements of the relevant subsurface properties are scarce and often of limited representativeness. The process of inferring subsurface properties from dependent information such as hydraulic head, chemical concentrations, or flow is known as inverse modelling (e.g., Jesús Carrera et al., 2005).

Unfortunately, as a consequence of the exceptional complexity of many hydrogeological systems (Figure 4-1), there usually exists more than a single plausible explanation for the observed data (Linde et al., 2015, 2017; Moeck et al., 2020). Variations in aquifer depth, sediment properties, atmospheric and hydrogeological forcing, anthropogenic influences, and complex geological features interact with each other and can create similar hydraulic responses in different arrangements. The consequence of this has been summarized succinctly by Poeter & Townsend (1994): "*A true evaluation of the possible subsurface configurations and their impact on the decision at hand is the only honest approach to groundwater analyses.*" and hence surmised that "*The era of drawing conclusions on the basis of deterministic flow and transport models has come to a close*".

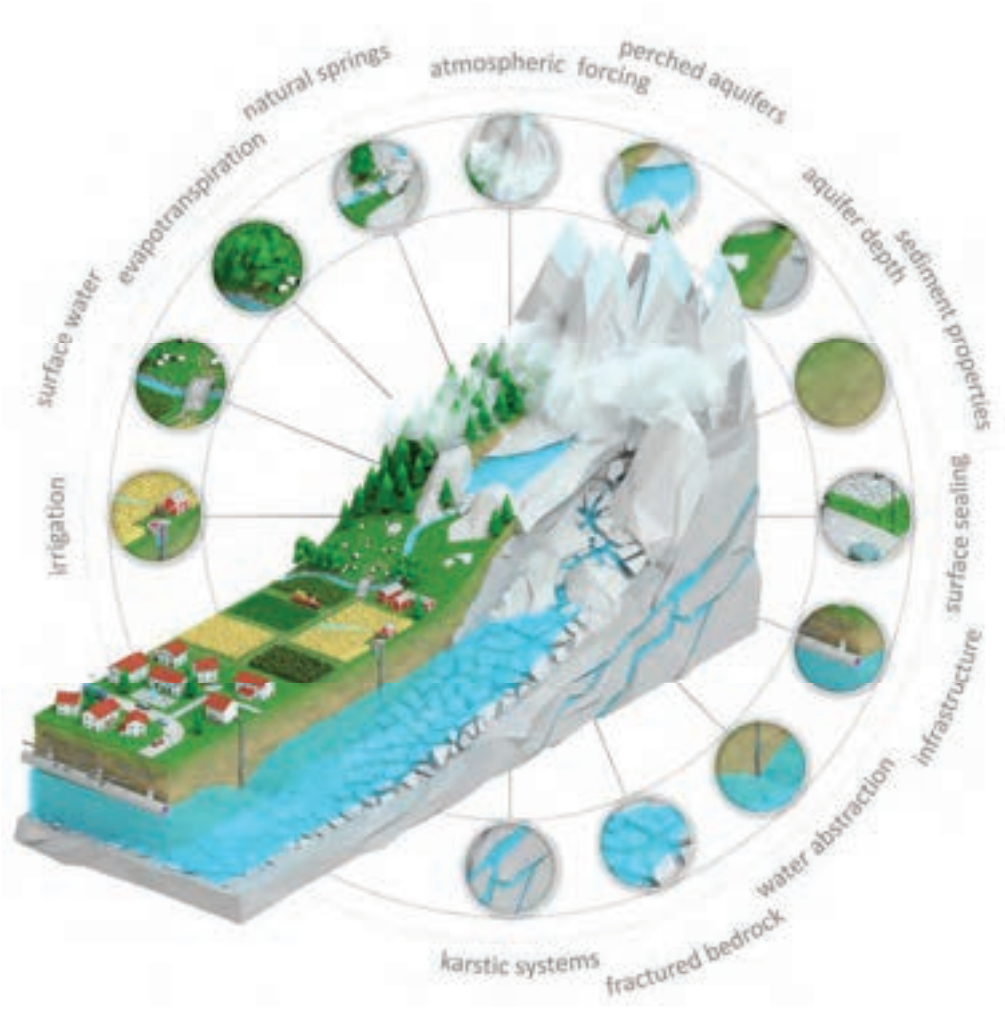


Figure 4-1. Complex and interacting aspects in a mountainous hydrogeological system. When the presence, properties, and extent of these aspects is not sufficiently quantified, they become sources of uncertainty for hydrogeological models.

Where deterministic models only seek a single promising model configuration, stochastic approaches based on Bayesian statistics explore multiple alternative configurations at once. This process hopes to identify ambiguities in order to endow model predictions with reliable uncertainty estimates. Unfortunately, 25 years later, Poeter & Townsend (1994)'s prediction has yet to fully come to pass. While the need for probabilistic groundwater models has been widely acknowledged (e.g., Cirpka & Valocchi, 2016; Renard, 2007; Sanchez-Vila & Fernàndez-Garcia, 2016), the complexity of representing the hydrogeological system

– and, by consequence, its uncertainties – remains an obstacle for the wide-scale adoption of Bayesian methods.

In Bayesian statistics, the plausibility of different narratives – as defined by model parameterizations – is represented through *probability density functions* (pdf). Bayes' theorem formalizes the synthesis of a so-called *posterior* from initial belief (the *prior*) and new data (the *likelihood*). Since it has no analytical solution in the general case, its practical use often demands approximations and simplifications. Among the most elegant is *Gaussianity*, which permits an analytical solution provided that the numerical model is linear, and that all pdfs involved are Gaussian. This assumption underlies the popular Ensemble Kalman Filter (EnKF: Evensen, 1994, 2003), which has proven easy to implement and highly robust to small ensemble sizes. As a consequence, it quickly gained popularity in the hydrogeological community (e.g., Gu & Oliver, 2007; Hendricks Franssen et al., 2011; Keller et al., 2018; Reichle et al., 2002). Unfortunately, the assumption of Gaussianity implies both unimodality (*there exists a single most probable solution*) and full support (*no solution is impossible*). Both assumptions are potentially problematic: the former because it cannot adequately represent the existence of distinct, equivalent solutions in the form of Pareto fronts or separate probability modes; the latter for parameters with strict physical limits.

It may seem expedient, then, to turn our attention to more general approaches such as *Markov Chain Monte Carlo* (MCMC: e.g., Foreman-Mackey et al., 2013; Smith & Marshall, 2008) or *particle filters* (PF: e.g., van Leeuwen, 2009; van Leeuwen et al., 2019). These methods can theoretically approximate arbitrary distributions but suffer from practical limitations of their own. Fundamentally, both methods suffer

in systems with high-dimensionality, although the specific symptoms vary: MCMC methods often display large auto-correlations if the proposal distributions are not sufficiently well-tuned, which reduces the sample generation efficiency significantly. Possible remedies are found in Hamiltonian Monte Carlo (e.g., Betancourt, 2018), which exploit Jacobian information, or the affine-invariant ensemble sampler for MCMC *emcee* (Foreman-Mackey et al., 2013), which restricts itself to a limited subspace. The ensembles underlying PFs, on the other hand, tend to quickly degenerate and collapse in high-dimensional systems (e.g., Arulampalam et al., 2002; Bengtsson et al., 2008), and may require pragmatic solutions which threaten to corrupt the inference (Moradkhani et al., 2005; Ramgraber et al., 2019; Vrugt et al., 2013). As such, these computational limitations render both methods less efficient in systems with limited computational resources than comparable Gaussian-based approaches.

In search of a free lunch, we would desire an inference algorithm which combines the strengths of the above: the efficiency and robustness of the EnKF in face of small ensemble sizes, and the PF's/MCMC's ability to explore non-Gaussian distributions. Stein Variational Gradient Descent (Liu & Wang, 2016), a relatively recent development in the computational sciences, may be an interesting step in this direction. Based on Kernelized Stein Discrepancy (Chwialkowski et al., 2016; Liu et al., 2016), it yields a surprisingly simple gradient descent algorithm capable of iteratively transforming an arbitrary ensemble of particles into samples of the posterior. With a few small adjustments, we shall see that it can share the EnKF's ability to scale the complexity of the inference problem by restricting the analysis to a parameter-subspace whose

dimensionality depends on the number of available particles, while at the same time being able to approximate non-Gaussian distributions. In the following, we will re-derive the algorithm, then propose adaptations and approximations required to render it tractable in practice. Afterwards, we will demonstrate its performance in a simple, bi-modal synthetic scenario, as well as in a highly complex pre-alpine catchment. Finally, we will discuss the results and provide an outlook for future research. First, however, we will present the nomenclature used in this study.

4.3 Theory

4.3.1 Nomenclature

In this study, we will use bold font to denote vectors or matrices and will refer to column vectors ($\boldsymbol{\theta} = [\theta_1, \dots, \theta_d]^\top$) unless otherwise specified. The symbol $\boldsymbol{\theta}$ denotes the vector of model parameters, and the variable $\boldsymbol{x}$ denotes model states. Data or observations are represented by $\boldsymbol{y}$. Standard font (e.g., θ) refers to scalar-valued variables. For functions, we shall refer to the function as an object by $\boldsymbol{f}$, and to its output by $\boldsymbol{f}(\boldsymbol{\theta})$. Functions with multiple arguments (e.g., $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$), for which one argument is assumed fixed, are denoted by a dot in its arguments (e.g., $k(\cdot, \boldsymbol{\theta}')$ for fixed $\boldsymbol{\theta}'$). $\|\boldsymbol{\theta}\|$ refers to the norm and $|\boldsymbol{\theta}|$ to the absolute value of $\boldsymbol{\theta}$. Superscripts in parentheses $\boldsymbol{\theta}^{(d)}$ refer to the d-th entry of $\boldsymbol{\theta}$. Capitalized roman normal symbols refer to integer variables: D to the dimensionality of parameter space (number of model parameters), O to the dimensionality of observation space (number of state observations), and N to the number of particles (ensemble size).

4.3.2 Stein Variational Gradient Descent

In the following, we will present the Stein Variational Gradient Descent (SVGD) algorithm following the procedure outlined in Liu et al. (2016) and Liu & Wang (2016). In short, SVGD iteratively transforms samples of an arbitrary reference distribution into samples of the posterior. This process may bear superficial similarity to filter techniques, but is based on a crucial difference: instead of sequentially adding information (think *treasure map*: specifying the steps to the target one by one), it homes in on the posterior distribution (think *navigation system*: constantly reorienting itself towards the target).

The algorithm is based on an incremental particle flow which iteratively transforms an ensemble of initial samples into posterior samples:

$$\boldsymbol{\theta_i} = \boldsymbol{T}(\boldsymbol{\theta_{i-1}}) = \boldsymbol{\theta_{i-1}} + \varepsilon \boldsymbol{\phi}^*(\boldsymbol{\theta_{i-1}}) \qquad 4\text{-}1$$

where the subscript i denotes the current iteration number, ε is a small scalar increment, and $\boldsymbol{\phi}^*: \mathbb{R}^D \to \mathbb{R}^D$ is a vector field whose pointwise evaluations $\boldsymbol{\phi}^*(\boldsymbol{\theta_{i-1}})$ designate the flow direction for each particle.

This vector field $\boldsymbol{\phi}^*$ is the key ingredient of SVGD. As we shall see in the following, it can be found through a function optimization on the space of vector fields/infinitesimal transformations. We identify the infinitesimal transformation that maximally reduces the Kullback-Leibler divergence (KLD) to the target posterior. The associated vector field thus corresponds to the negative functional gradient of the KLD, and its norm defines a discrepancy measure called the Kernelized Stein Discrepancy (KSD). The resulting equation for $\boldsymbol{\phi}^*$ is surprisingly simple, providing the particle flow directions for an infinitesimally small step towards the posterior distribution.

However, in order to understand the derivation of the algorithm, we must introduce the concept of a *Reproducing Kernel Hilbert Space* (*RKHS*).

4.3.2.1 Reproducing Kernel Hilbert Spaces

RKHS are special, infinite-dimensional function spaces with a number of properties which make them interesting for functional optimization tasks – tasks, in which we want to find functions which fulfil certain requirements.

There are several alternative ways to define a RKHS $\mathcal{H}$. In this study, we adopt the definition used in Liu et al. (2016). This definition is based on the spectral decomposition of a positive definite, symmetric kernel $k(\boldsymbol{\theta}, \boldsymbol{\theta}'): \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. An example of such a kernel is the *radial basis function* (RBF) kernel:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp\left(-\frac{\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2}{2h^2}\right) \qquad 4\text{-}2$$

where h^2 is the kernel's bandwidth. These kernels can be thought of as similarity metrics between two particles $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$: if the particles are identical, the kernel yields 1, and the more different they are, the closer the kernel's output will be to zero. According to *Mercer's theorem*, any symmetric, positive semi-definite kernel is associated with an inner product on some Hilbert space $\mathcal{H}$, obtained through spectral decomposition of the Hilbert-Schmidt integral operator (e.g., Schölkopf & Smola, 2001; D. Werner, 2018):

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{l=1}^{\infty} \lambda_l e_l(\boldsymbol{\theta}) e_l(\boldsymbol{\theta}') \qquad 4\text{-}3$$

This expresses the kernel as an infinite series of orthonormal eigenfunctions e_l and eigenvalues λ_l. These eigenfunctions can be

interpreted as an orthonormal basis which spans up an infinite-dimensional RKHS $\mathcal{H}$ which comprises of linear combinations of its eigenfunctions $f(\boldsymbol{\theta}) = \sum_{l=1}^{\infty} f_l e_l(\boldsymbol{\theta})$ with $\sum_{l=1}^{\infty} {f_l}^2/\lambda_l < \infty$ and an inner product $\langle f, g \rangle_{\mathcal{H}} = \sum_{l=1}^{\infty} f_l g_l/\lambda_l$ between $f(\boldsymbol{\theta})$ and $g(\boldsymbol{\theta}) = \sum_{l=1}^{\infty} g_l e_l(\boldsymbol{\theta})$. This also defines a norm $\|f\|_{\mathcal{H}}$ where $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}} = \sum_{l=1}^{\infty} {f_l}^2/\lambda_l$.

Equation 4-3 may then be interpreted as an inner product between two vectors $\boldsymbol{k}(\cdot, \boldsymbol{\theta})$ and $\boldsymbol{k}(\cdot, \boldsymbol{\theta}')$ in $\mathcal{H}$. Since their embedding space $\mathcal{H}$ is infinite-dimensional, these vectors will have infinitely many entries:

$$\boldsymbol{k}(\boldsymbol{\theta}, \cdot) = \left[\sqrt{\lambda_1} e_1(\boldsymbol{\theta}), \dots, \sqrt{\lambda_\infty} e_\infty(\boldsymbol{\theta})\right]^\top \qquad 4\text{-}4$$

$$\boldsymbol{k}(\cdot, \boldsymbol{\theta}') = \left[\sqrt{\lambda_1} e_1(\boldsymbol{\theta}'), \dots, \sqrt{\lambda_\infty} e_\infty(\boldsymbol{\theta}')\right]^\top \qquad 4\text{-}5$$

Why is this useful? In machine learning literature, particularly for classification tasks (e.g., *support vector machines*: Schölkopf & Smola, 2001), it is common to extract *features* (here: $\sqrt{\lambda_l} e_l(\boldsymbol{\theta})$) from an *input data set* (here: $\boldsymbol{\theta}$). The larger the amount of independent, extracted features, the easier the classification becomes. In a RKHS, the number of these features is infinite. And if the only operation on these features required is an inner product, we need not even compute them – an evaluation of the kernel would yield the desired result. We can verify that an inner product between Equation 4-4 and Equation 4-5 yields Equation 4-3, and retrieve one of the fundamental properties of a RKHS $\mathcal{H}$:

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \langle \boldsymbol{k}(\boldsymbol{\theta}, \cdot), \boldsymbol{k}(\cdot, \boldsymbol{\theta}') \rangle_{\mathcal{H}} = \sum_{l=1}^{\infty} \sqrt{\lambda_l} e_l(\boldsymbol{\theta}) \sqrt{\lambda_l} e_l(\boldsymbol{\theta}') = \sum_{l=1}^{\infty} \lambda_l e_l(\boldsymbol{\theta}) e_l(\boldsymbol{\theta}') \qquad 4\text{-}6$$

For the purpose of functional optimization, we are interested in the functions defined in the RKHS. $\mathcal{H}$ contains scalar-valued functions f mapping from the parameter space ($f: \mathbb{R}^D \to \mathbb{R}$) which are constructed through linear combinations of its basis, the eigenfunctions:

$$f(\boldsymbol{\theta}) = \sum_{l=1}^{\infty} f_l e_l(\boldsymbol{\theta}) \qquad 4\text{-}7$$

where f_l are some arbitrary real scalars. These functions are uniquely defined by a vector $\boldsymbol{f}(\cdot)$ in $\mathcal{H}$

$$\boldsymbol{f}(\cdot) = \left[f_1/\sqrt{\lambda_1}, \dots, f_\infty/\sqrt{\lambda_\infty}\right]^T \qquad 4\text{-}8$$

and can be retrieved by taking an inner product with Equation 4-5 (replacing $\boldsymbol{\theta}'$ with $\boldsymbol{\theta}$). This defines the RKHS's eponymous *reproducing property*:

$$f(\boldsymbol{\theta}) = \langle \boldsymbol{f}(\cdot), \boldsymbol{k}(\cdot, \boldsymbol{\theta}) \rangle_{\mathcal{H}} = \sum_{l=1}^{\infty} \frac{f_l}{\sqrt{\lambda_l}} \sqrt{\lambda_l} e_l(\boldsymbol{\theta}) = \sum_{l=1}^{\infty} f_l e_l(\boldsymbol{\theta}) \qquad 4\text{-}9$$

With the fundamentals of RKHS defined, let us proceed to the derivation of the algorithm.

4.3.2.2 Deriving the algorithm

SVGD is derived from a metric called *Kernelized Stein Discrepancy* (*KSD*: Chwialkowski et al., 2016; Liu et al., 2016) $\mathbb{S}(q||p)$ between two probability distributions q and p. This metric yields a measure of discrepancy between the two distributions, provided that we have an ensemble of samples from q and are able to evaluate the gradient of the logarithm of p at least pointwise. In our application, q will always be some intermediate distribution from which we assume our samples are drawn, and p will be the target posterior.

$$\mathbb{S}(q||p) = \max_{\boldsymbol{\phi} \in \mathcal{F}} \left\{ \left[\mathbb{E}_{\boldsymbol{\theta} \sim q} \left[\operatorname{trace} \boldsymbol{A}_p \boldsymbol{\phi}(\boldsymbol{\theta}) \right] \right]^2 \right\} \qquad 4\text{-}10$$

In Equation 4-10, $\mathbb{E}_{\boldsymbol{\theta} \sim q}$ refers to the expectation under the assumption that the particles $\boldsymbol{\theta}$ are sampled from q, $\boldsymbol{\phi}$ is a vector field on parameter

space, representing an infinitesimal transformation, and $\boldsymbol{A}_p$ is a linear operator:

$$\boldsymbol{A}_p\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{\phi}(\boldsymbol{\theta})[\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta})]^\top + \boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{\phi}(\boldsymbol{\theta}) \qquad 4\text{-}11$$

where $\boldsymbol{\nabla}_{\boldsymbol{\theta}} = \left[\partial/\partial\boldsymbol{\theta}^{(1)}, \dots, \partial/\partial\boldsymbol{\theta}^{(1)}\right]^\top$ denotes the partial derivative operator evaluated at $\boldsymbol{\theta}$. We have provided a detailed derivation of Equation 4-10 in *Appendix 1: Kernelized Stein Discrepancy*. The challenging part in Equation 4-10 is the functional optimization, that is the need to find the vector field $\boldsymbol{\phi}^*$ which maximizes the violation of Stein's identity. Fortunately, this is where the properties of the RKHS prove advantageous. If we assume the family of functions $\mathcal{F}$ are the functions we can define in a RKHS (Equation 4-8 and 4-9), the functional optimization in Equation 4-10 has a closed-form solution:

$$\boldsymbol{\phi}^*(\boldsymbol{\theta}') = \mathbb{E}_{\boldsymbol{\theta}\sim q}[k(\boldsymbol{\theta}, \boldsymbol{\theta}')\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \boldsymbol{\nabla}_{\boldsymbol{\theta}} k(\boldsymbol{\theta}, \boldsymbol{\theta}')] \qquad 4\text{-}12$$

We have re-derived this solution in detail in *Appendix 2: Functional optimization in KSD*. The vector-valued function $\boldsymbol{\phi}^*$ defines a vector field over the parameter space $\mathbb{R}^D$, and assigns to each position a D-dimensional vector or direction which maximizes the violation of Stein's identity.

SVGD exploits this information to implement a particle flow which gradually transforms the distribution q into the distribution p, the posterior. It can be shown (Liu & Wang, 2016), that for linear invertible transformations the directions $\boldsymbol{\phi}^*(\boldsymbol{\theta}')$ of the vector field $\boldsymbol{\phi}^*$ correspond to the steepest descent directions of the *Kullback-Leibler divergence* (KLD). We have re-derived this for the reader's convenience in *Appendix 3: Relation to KLD*. Using the transformation in Equation **4-1**, we establish an iterative particle flow through parameter space. The steepest descent

directions $\boldsymbol{\phi}^*(\boldsymbol{\theta})$ are obtained by taking an ensemble approximation of Equation 4-12:

$$\boldsymbol{\phi}^*(\boldsymbol{\theta}) = \frac{1}{N}\sum_{j=1}^{N} k(\boldsymbol{\theta_j}, \boldsymbol{\theta})\nabla_{\boldsymbol{\theta_j}} \log p(\boldsymbol{\theta_j}) + \nabla_{\boldsymbol{\theta_j}} k(\boldsymbol{\theta_j}, \boldsymbol{\theta}) \qquad 4\text{-}13$$

The only really expensive variable to evaluate is the gradient of the logposterior at the particle positions $\nabla_{\boldsymbol{\theta_j}} \log p(\boldsymbol{\theta_j})$. In general cases, where no analytic form for the logposterior or its derivative are available, we must resort to approximations of this gradient. We will investigate a few approaches towards this end in the following section.

4.4 Algorithmic approximations

4.4.1 Posterior gradient $\nabla_\theta \log p(\theta)$

To evaluate $\nabla_{\boldsymbol{\theta_j}} \log p(\boldsymbol{\theta_j})$, we follow the approach presented by Pulido et al. (2019). We start with Bayes' theorem:

$$f(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f(\boldsymbol{\theta})f(\boldsymbol{y}|\boldsymbol{\theta})}{f(\boldsymbol{y})} \qquad 4\text{-}14$$

where $f(\boldsymbol{\theta}|\boldsymbol{y}) \coloneqq p(\boldsymbol{\theta})$ is the posterior pdf, $f(\boldsymbol{\theta})$ the prior pdf, $f(\boldsymbol{y}|\boldsymbol{\theta})$ the likelihood, and $f(\boldsymbol{y})$ the model evidence. If we now apply a logarithm to both sides, then a partial derivative operator, and expand the fraction within the logarithm's argument, we obtain:

$$\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}|\boldsymbol{y}) = \nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{y}|\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{y}) \qquad 4\text{-}15$$

Acknowledging that $f(\boldsymbol{y})$ is a constant with respect to $\boldsymbol{\theta}$, it follows that $\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{y}) = 0$. This is also listed as an initial motivation of SVGD: that the gradient of the logposterior can be calculated without knowing the model evidence $f(\boldsymbol{y})$. Consequently, we can calculate the gradient of the

logposterior from the gradient of the logprior and the gradient of the loglikelihood alone:

$$\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}|\boldsymbol{y}) = \nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}, \boldsymbol{y}) = \nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{y}|\boldsymbol{\theta}) \quad 4\text{-}16$$

Obtaining $\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{\theta})$ is relatively straightforward, as the definition of the prior is left to us. As such, we can focus on $\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{y}|\boldsymbol{\theta})$. If we assume multivariate Gaussian likelihoods, we have:

$$f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^O \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{y}_{sim})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{y}_{sim})\right) \quad 4\text{-}17$$

where O is the number of observations or the dimensionality of observation space, $\boldsymbol{\Sigma}^{-1}$ is the inverse of the covariance matrix, and $\boldsymbol{y}_{obs}$ and $\boldsymbol{y}_{sim}$ refer to the observed and simulated states. If we first take the logarithm and then the partial derivatives, we obtain:

$$\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{2} \nabla_{\boldsymbol{x}} \boldsymbol{y}_{sim}{}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{y}_{sim}) \quad 4\text{-}18$$

If we simulate the states with a numerical model which takes as input parameters $\boldsymbol{x}$, i.e.:

$$\boldsymbol{y}_{sim} = \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}) \quad 4\text{-}19$$

where we simplified notation slightly by implying that the model simulates the observed states directly. In practice, the model would generate the full state space (i.e., a time series of water table fields), and we would extract only the relevant entries – for example at the locations of observation wells at certain times. We can plug this into Equation 4-18:

$$\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{2} \nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}) \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{y}_{obs} - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})\right) \quad 4\text{-}20$$

Or, re-phrased in terms of the local Jacobian $\boldsymbol{J}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})^\top$:

$$\nabla_{\boldsymbol{\theta}} \log f(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{J}(\boldsymbol{\theta})^{\top}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{y}_{obs} - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})\right) \quad 4\text{-}21$$

4.4.2 Jacobian matrix $\nabla_{\theta}\mathcal{M}(\theta)$

The computational bottleneck for the solution of Equation 4-20 are the model predictions $\boldsymbol{\mathcal{M}}(\boldsymbol{\theta})$, an $O \times 1$ vector, and by extension the local Jacobian $\boldsymbol{J}(\boldsymbol{\theta})$, an $O \times D$ matrix. While the former can be obtained by simply running the model, the latter is not generally available in closed form. Some recent developments like automatic differentiation (e.g., Margossian, 2019) hold promise for future applications, but are model-intrusive and not yet widely supported.

Instead, it may be necessary to apply non-intrusive approximations of the Jacobian. The standard numerical approach consists of perturbing the parameter vector $\boldsymbol{\theta}$ by a small increment along each dimension, then filling the Jacobian matrix with the resulting two-point (or three-point) finite difference derivatives (e.g., Wendt et al., 2009). While this numerical differentiation can yield very precise approximations, it quickly becomes computationally unfeasible: To obtain the set of local Jacobians, we would have to run the model $N(D+1)$ times (or $N(2D+1)$ times for three-point derivatives) in each iteration. For complex, computationally demanding models, we generally cannot afford more than N model evaluations, since the ensemble size is usually adjusted to the computational resources available.

As such, we would like to estimate the Jacobian directly from the ensemble, using only the N model evaluations $\boldsymbol{\mathcal{M}}(\boldsymbol{\theta})$. One such approach has been used by Chen & Oliver (2013) and White (2018), approximating the Jacobian based on each ensemble member's deviation from the mean. This approach is very powerful, but is unfortunately

based on the assumption of Gaussianity, and thus squanders the non-Gaussian inference which motivated the use of SVGD in the first place.

Pulido et al. (2019) suggest an alternative approach, which consists of defining the observation operator (analogous to our model $\boldsymbol{\mathcal{M}}$) in a RKHS, then shifting the derivative operator to the kernel:

$$\boldsymbol{J}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \boldsymbol{\mathcal{M}}(\boldsymbol{\theta_n}) \boldsymbol{\nabla_{\theta}} k(\boldsymbol{\theta}, \boldsymbol{\theta_n}) \qquad 4\text{-}22$$

This approach can also be interpreted as an approximation of $\boldsymbol{\mathcal{M}}$ with radial basis functions, then analytically evaluating the derivatives of this approximation. While this approximation yields favorable results in certain cases, it does not generalize very well to groundwater models. First and foremost, if we evaluate Equation 4-22 at a particle's position, the derivative is informed exclusively by other particles. This is because the kernel derivative evaluated at its own centre is zero ($\boldsymbol{\nabla_{\theta}} k(\boldsymbol{\theta}, \boldsymbol{\theta}) = 0$). A further property of this RBF approximation is that its limits tend towards zero ($\lim_{\boldsymbol{\theta} \to \infty} k(\boldsymbol{\theta}, \boldsymbol{\theta_n}) = 0$) and the derivative therefore depends on the model output's absolute magnitude and sign. For hydrogeological states – which are often defined in m above sea level –, this is clearly an undesirable property. Consequently, we employ a different ensemble-based approximation, endeavouring to retain the localization of Pulido et al. (2019)'s approach while exploiting relative differences to render the method independent of absolute magnitudes.

Our approach is best presented as an approximate, reverse *singular value decomposition* (SVD). Taking the SVD of a Jacobian would yield three matrices $\boldsymbol{U}$, $\boldsymbol{S}$, and $\boldsymbol{V}^{\top}$, where $\boldsymbol{U}$ is a $O \times O$ matrix containing the left singular vectors $\boldsymbol{u}_i, i = 1, \ldots, O$, corresponding to a new, reoriented basis

of state space dimensions, $\boldsymbol{S}$ is a diagonal $O \times D$ matrix containing the sorted singular values $s_i, i = 1, \ldots, S$, of the Jacobian, corresponding to the absolute gradients along the new basis of the singular vectors, and $\boldsymbol{V}^\top$ is a transpose $D \times D$ matrix of the right singular vectors $\boldsymbol{v}_i, i = 1, \ldots, D$, corresponding to a new, reoriented basis of parameter space dimensions:

$$\boldsymbol{J} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top \tag{4-23}$$

This form reveals interesting properties of the Jacobian, because if the central matrix $\boldsymbol{S}$ is not square or of full rank, some parameter and/or state dimensions lie in a so-called null space, about which the Jacobian contains no information. Due to the orthogonality of the singular vectors, we could break up this form further, splitting it into a set of one-dimensional gradient matrices, composed of the dot product between the i-th left singular vector $\boldsymbol{u}_i$, scalar value s_i, and transpose right singular vector $\boldsymbol{v}_i{}^\top$.

$$\boldsymbol{J} = \sum_{i=1}^{S} \boldsymbol{J}_i = \sum_{i=1}^{S} \boldsymbol{u}_i s_i \boldsymbol{v}_i{}^\top \tag{4-24}$$

where $S \leq D$. The Jacobian form in Equation 4-24 allows for an interesting interpretation: If we were to evaluate the model a small distance ε along the right singular vectors $\boldsymbol{v}_i$, we would elicit a model response along $\boldsymbol{u}_i$. The singular vector s_i, then, corresponds to the scalar gradient norm of the response.

$$\boldsymbol{u}_i = \frac{\boldsymbol{\mathcal{M}}(\boldsymbol{\theta} + \varepsilon \boldsymbol{v}_i) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})}{\|\boldsymbol{\mathcal{M}}(\boldsymbol{\theta} + \varepsilon \boldsymbol{v}_i) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})\|} \tag{4-25}$$

$$s_i = \frac{\|\boldsymbol{\mathcal{M}}(\boldsymbol{\theta} + \varepsilon \boldsymbol{v}_i) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})\|}{\|\boldsymbol{\theta} + \varepsilon \boldsymbol{v}_i - \boldsymbol{\theta}\|} = \frac{\|\boldsymbol{\mathcal{M}}(\boldsymbol{\theta} + \varepsilon \boldsymbol{v}_i) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})\|}{\|\varepsilon \boldsymbol{v}_i\|} = \frac{\|\boldsymbol{\mathcal{M}}(\boldsymbol{\theta} + \varepsilon \boldsymbol{v}_i) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta})\|}{|\varepsilon|} \quad 4\text{-}26$$

We could use this information to reconstruct the full Jacobian with S model evaluations, or an incomplete Jacobian if we have only access to some but not all right singular vectors. In practice, we only have access to the vectors between pairs of particles. To obtain a Jacobian estimate at particle $\boldsymbol{\theta}_n$, we can approximate gradient matrices

$$\tilde{\boldsymbol{J}}_m = \tilde{\boldsymbol{u}}_m g_m \tilde{\boldsymbol{v}}_m{}^\top \quad 4\text{-}27$$

for $m = 1, \dots, N$ with

$$\tilde{\boldsymbol{u}}_m = \frac{\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)}{\|\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\|} \quad 4\text{-}28$$

$$g_m = \frac{\|\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\|}{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|} \quad 4\text{-}29$$

$$\tilde{\boldsymbol{v}}_m{}^\top = \frac{\boldsymbol{\theta}_m{}^\top - \boldsymbol{\theta}_n{}^\top}{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|} \quad 4\text{-}30$$

In pursuit of a Jacobian approximation, it is clear that simply summing the gradient matrices $\tilde{\boldsymbol{J}}_m$ will not yield the correct results: If $N > D$, the magnitude of the resulting Jacobian's gradients will be too large. This is because the gradient matrices can no longer be orthogonal to each other (it is impossible to have more independent vectors than space dimensions) and will thus accumulate redundant gradient components. Taking a simple arithmetic average, on the other hand, also bears error potential: if we consider the case of $N = S$ and $\{\tilde{\boldsymbol{J}}_m\}_{m=1,\dots,N} = \{\boldsymbol{J}_i\}_{i=1,\dots,S}$, the resulting Jacobian estimate would be underestimated by a factor of

S. This is because every gradient matrix $\tilde{\boldsymbol{J}}_m$ is only of rank 1, and so contributes at most one dimension to a rank S Jacobian (Equation 4-24).

We might be tempted to multiply the average by the dimensionality of parameter space D to remediate this issue. This is an improvement, but obviously not perfect. For its limitations, consider the following scenario: if the particle positions happen to be systematically biased, or multiple particles would be located in a geometrically degenerate way, the Jacobian's gradients along one dimension could be severely overestimated. Consider the case $D = 3, N = 10$. If all 10 particles were positioned along a straight line (i.e., restricted to a one-dimensional subspace), the gradient along this line would be overestimated by a factor of 3 (but fortunately not 10, thanks to the factor $\frac{D}{N}$). This is because a flat correction by a factor of D does not account for the possibility of geometrical degeneracy. We could partially prevent such scenarios by estimating the dimensionality V of the span of the vectors before (for example through Gram-Schmidt-Orthogonalization or principle component analysis), then selecting a scaling factor P as the smallest integer among D, $N-1$, and V:

$$P = \min\{D, N-1, V\} \tag{4-31}$$

Using this scaling factor, we obtain:

$$\tilde{\boldsymbol{J}}(\boldsymbol{\theta}_n) = \frac{P}{N}\sum_{m=1}^{N}\tilde{\boldsymbol{J}}_m = \frac{P}{N}\sum_{m=1}^{N}\left(\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\right)\frac{(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^\top}{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|^2} \tag{4-32}$$

For further intuition, we could interpret Equation 4-32 in terms of deviation vectors $\boldsymbol{\vartheta}_m = \boldsymbol{\theta}_m - \boldsymbol{\theta}_n$, which would yield:

$$\tilde{\boldsymbol{J}}(\boldsymbol{\theta}_n) = \frac{P}{N}\sum_{m=1}^{N}\left(\boldsymbol{\mathcal{M}}(\boldsymbol{\theta_n} + \boldsymbol{\vartheta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\right)\frac{\boldsymbol{\vartheta}_m{}^{\top}}{\|\boldsymbol{\vartheta}_m\|^2} \qquad 4\text{-}33$$

We expect that if these deviation vectors can be assumed to be i.i.d. samples from a standard normal distribution, then for linear functions (which feature constant Jacobians), $\tilde{\boldsymbol{J}}(\boldsymbol{\theta}_n)$ would converge against the real $\boldsymbol{J}(\boldsymbol{\theta}_n)$ in the limit of $N \to \infty$.[14]

Unfortunately, in practice we often deal with nonlinear functions. In order to justify a linear approximation, we might want to limit the influence of distant particles on the local Jacobian approximation. As a consequence, it seems expedient to replace the arithmetic average in Equation 4-32 with distance-based weights, for example based on kernel contributions:

$$w_m = k_{\boldsymbol{\theta}_n}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_m) \Big/ \sum\nolimits_{l \neq n} k_{\boldsymbol{\theta}_n}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_l) \qquad 4\text{-}34$$

$$\tilde{\boldsymbol{J}}(\boldsymbol{\theta}_n) = P\sum_{m=1}^{N} w_m\left(\boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_m) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)\right)\frac{(\boldsymbol{\theta}_m - \boldsymbol{\theta}_n)^{\top}}{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}_n\|^2} \qquad 4\text{-}35$$

where $k_{\boldsymbol{\theta}_n}$ is a radial basis kernel (Equation 4-2) whose bandwidth is, for example, based on the median distance of the other particles to $\boldsymbol{\theta}_n$. Since the state spaces in our scenarios are at least as large as the parameter spaces, and our samples i.i.d., we do not expect geometric degeneracy. As such, we fix P at D or $N - 1$, whichever is smaller. Pseudo-code for the Jacobian approximation is provided in Figure 4-2.

[14] In case of directional bias of the particle (and consequently deviation vector $\boldsymbol{\vartheta}_m$) arrangement, this bias will likely propagate to the Jacobian estimate.

Step 1: Create empty Jacobian

$J(\boldsymbol{\theta}_n) = zeros(O \times D)$

For particle m **from** 1 **to** N, **if** $m \neq n$**:**

Step 2: Create difference vectors

$\boldsymbol{u}_m = \boldsymbol{\mathcal{M}}(\boldsymbol{\theta_m}) - \boldsymbol{\mathcal{M}}(\boldsymbol{\theta_n})$

$\boldsymbol{v}_m = \boldsymbol{\theta}_m - \boldsymbol{\theta}_m$

Step 3: Create their normalized variants

$\tilde{\boldsymbol{u}}_m = \boldsymbol{u}_m / \|\boldsymbol{u}_m\|$

$\tilde{\boldsymbol{v}}_m = \boldsymbol{v}_m / \|\boldsymbol{v}_m\|$

Step 4: Calculate the scalar gradient

$g_m = \|\boldsymbol{u}_m\| / \|\boldsymbol{v}_m\|$

Step 5: Determine gradient matrix

$\tilde{\boldsymbol{J}}_m = \tilde{\boldsymbol{u}}_m g_m \tilde{\boldsymbol{v}}_m{}^{\top}$ (see Equation 4-27)

Step 6: *Find individual kernel bandwidth*

1. Set h of $k_{\boldsymbol{\theta_n}}$ to k-th median distance to other particles

Step 7: Calculate kernel weight

$w = k_{\boldsymbol{\theta}_n}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_m) / \sum_{l \neq n} k_{\boldsymbol{\theta}_n}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_l)$ (see Equation 4-34)

Step 8: Add contribution to Jacobian

$\boldsymbol{J}(\boldsymbol{\theta}_n) = \boldsymbol{J}(\boldsymbol{\theta}_n) + w\tilde{\boldsymbol{J}}_m$

Figure 4-2. Pseudo-code for the Jacobian approximation used in this study. Without additional model runs, evaluations of the Jacobian are only possible at the particle positions.

4.4.3 Gradient Descent algorithm

For an efficient inference with SVGD, we do not only require the descent directions $\boldsymbol{\phi}^*(\boldsymbol{\theta})$, but also an adaptive scheme to adjust the step-size ε. If the step-size is too small, the algorithm may require too many iterations to be computationally feasible. If the step-size is too large, the algorithm may overshoot, start oscillating, and fail to locate the optimum at all. As such, we would like to adjust the step-size dynamically.

Many such algorithms exist. Methods like *adaptive moment estimation* (ADAM: Kingma & Ba, 2015) or *adaptive subgradient methods* (AdaGrad: Duchi et al., 2011) have proven successful in the optimization of machine

learning algorithms, being capable of dynamically adjusting the gradient descent to improve efficiency. Unfortunately, they often employ individual step-sizes for each parameter space dimension or otherwise alter the gradient vectors at each position through momentum. Since the theory derived above assumes a scalar, uniform ε at each iteration, we construct an alternative descent algorithm for this study:

$$a_{i,n} = \alpha^{\left(\left\langle \frac{\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})}{\|\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})\|}, \frac{\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})}{\|\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})\|} \right\rangle - \beta\right)} \min\left(1, \frac{\|\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})\|}{\|\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})\|}\right) \quad 4\text{-}36$$

$$\varepsilon_i = \varepsilon_{i-1} \frac{1}{N} \sum_{n=1}^{N} a_{i,n} \quad 4\text{-}37$$

This step-size update algorithm does not affect the gradient in any way, but may require some explanation to become intuitive. It requires two hyperparameters: an acceleration rate $\alpha > 1$, and a similarity cutoff $0 \leq \beta < 1$. At each iteration, the previous step-size ε_{i-1} is rescaled by a factor (Equation 4-37) corresponding to the ensemble mean of all acceleration proposals $a_{i,n}$ (Equation 4-36). These acceleration proposals are composed of two terms: the first term compares the directions of two subsequent descent vectors, proposing acceleration if the directions are sufficiently similar and deceleration if they are not; the second term compares the norm of two subsequent descent vectors, proposing deceleration if the norm (and thus velocity) of the vector increases.[15]

[15] For further intuition, imagine an automatic car driving along the bottom of a winding valley. We would like to prescribe rules which speed up the voyage while guaranteeing a minimum of safety. One rule we might ascribe could be that whenever the car finds itself on a straight road, it should accelerate. Conversely, if the car approaches a curve, it should slow down. This is what the first term does. If the road further dips downward and would thus accelerate due to the slope, we should reduce the acceleration accordingly (or even break) to maintain a controllable velocity. This is what the second term does.

For the first part, we exponentiate α by the inner product between the normalized current descent direction $\frac{\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})}{\|\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})\|}$ and the normalized previous descent direction $\frac{\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})}{\|\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})\|}$. This compares the similarity of both vectors and accelerates or slows the descent accordingly. Since a naïve inner product would only stop accelerating for turns sharper than 90° – and we may want to stop accelerating long before that – the second hyperparameter β is subtracted from the inner product. A cutoff of $\beta = 0.75$, for example, restricts acceleration to a cone of about 40° around the previous vector. For the second part, if the norm of the descent algorithm is increasing ($\|\boldsymbol{\phi}_{i-1}^*(\boldsymbol{\theta}_{i-1,n})\| < \|\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_{i,n})\|$), the step-size should be reduced proportionally, to reduce the risk of shooting past the optimum if the descent direction remains the same.

4.4.4 Pseudocode

To summarize the algorithmic approximations used in this study, pseudo-code for the algorithm is provided in Figure 4-3.

Initialization

- Draw N particles from the prior $f(\boldsymbol{\theta})$
- Generate (or draw) the initial states $\boldsymbol{x}_0$
- Define initial stepsize ε_0, acceleration rate α, and cutoff β
- Set iteration counter $i = 0$ and iteration Boolean $iterate = True$

While $iterate = True$**:**

$i \to i + 1$

Step 1: Numerical simulation

For particle n **from** 1 **to** N:

1. Use pre-processors to calculate auxiliary variables (if applicable)
2. Simulate and extract observed variables $\boldsymbol{x}_n = \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n, \boldsymbol{x}_0, \dots)$

Step 2: Determine Gram matrix and kernel derivatives

For each particle index pair $(n, m) \in \{1, \dots, N\} \times \{1, \dots, N\}$:

1. Evaluate the kernel $k(\boldsymbol{\theta}_n, \boldsymbol{\theta}_m)$
2. Evaluate the kernel gradient $\boldsymbol{\nabla}_{\boldsymbol{\theta}_n} k(\boldsymbol{\theta}_n, \boldsymbol{\theta}_m)$

Step 3: Approximate ensemble-based Jacobian

For particle n **from** 1 **to** N:

1. Calculate $\boldsymbol{\nabla}_{\boldsymbol{\theta}_n} \boldsymbol{\mathcal{M}}(\boldsymbol{\theta}_n)$ (see Chapter 4.4.2)

Step 4: Determine gradient descent direction

For particle n **from** 1 **to** N:

1. Calculate direction $\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)$ (see Equation 4-13)
2. Normalize it to obtain $\overline{\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)}$

Step 5: Identify gradient similarity (see Chapter 4.4.3)

If $i > 1$:

For particle n **from** 1 **to** N:

1. Calculate acceleration proposal $a_{i,n}$ (see Equation 4-36)

Step 6: Adjust gradient descent step size

1. Average acceleration proposals to get $\bar{a}_i$
2. Adjust step size $\varepsilon_i = \varepsilon_{i-1} \bar{a}_i$ (see Equation 4-37)

Step 7: Apply gradient descent

For particle n **from** 1 to N:

1. Check for limits of $\boldsymbol{\theta}_{i,n} + \varepsilon_i \boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)$, adjust $\boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)$ if required
2. Update particles $\boldsymbol{\theta}_{i+1,n} = \boldsymbol{\theta}_{i,n} + \varepsilon_i \boldsymbol{\phi}_i^*(\boldsymbol{\theta}_n)$ (see Equation 4-1)

Step 8: Check for convergence

If convergence criterium fulfilled:

Set $iterate = False$

Figure 4-3. Pseudo-code of the SVGD algorithm used in this study. Step 3 can be replaced if other methods for obtaining the Jacobian are available, Steps 4.2 to Step 6 may be replaced if a different Gradient Descent method is used.

4.5 Synthetic test case

4.5.1 Setup

To illustrate the practical capabilities of SVGD, we first consider a simple synthetic test case. Towards this end, we construct a numerical hydrogeological model with a single parameter informing the uncertain path of a high-conductive paleo-channel in a two-dimensional, unconfined setting. This setup is illustrated in Figure 4-4. The system is defined as steady-state. Flow is driven by uniform recharge of 10^{-6} m/s over the model domain and drains to the southern fixed-head border. All other borders are assumed no-flow. Hydraulic conductivities of the

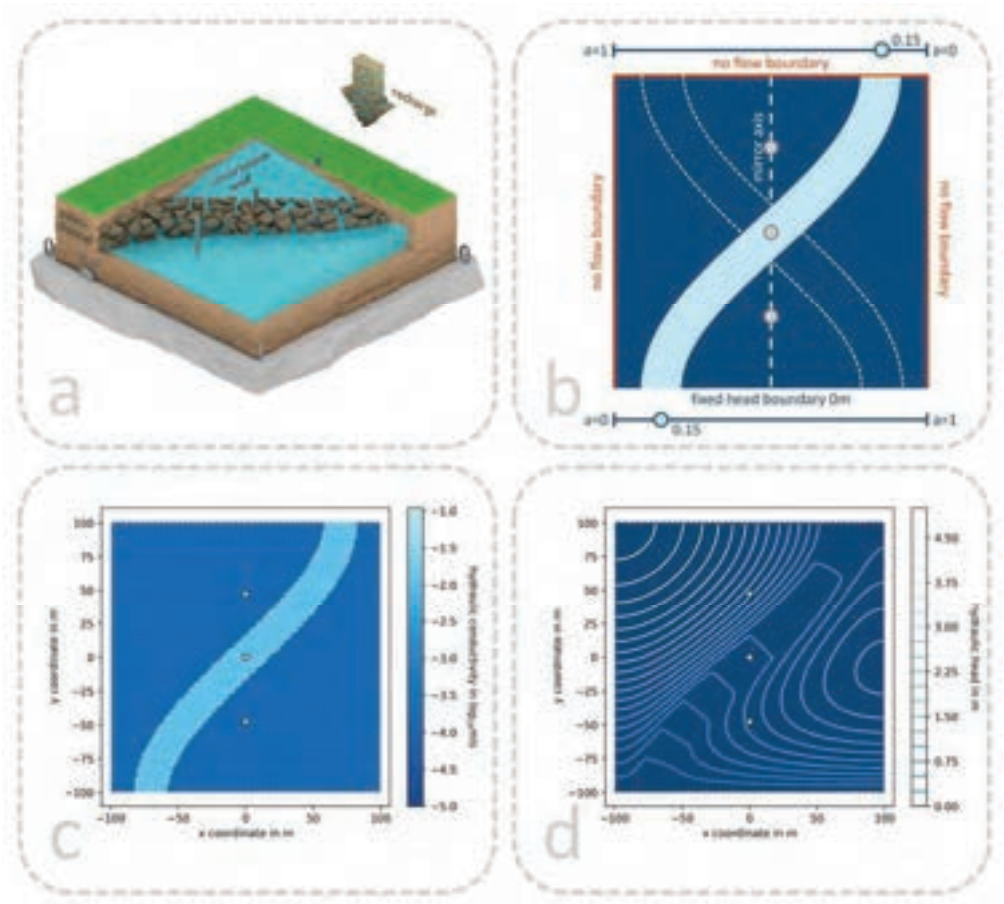


Figure 4-4. Conceptual render (a), conceptual sketch (b), true hydraulic conductivity field (c), and resulting true hydraulic head field (d) of the simple synthetic test case.

background and paleo-channel are defined as 10^{-4} m/s and 10^{-2} m/s, respectively. Specific yield was set to $S_y = 0.15$, and the top and bottom elevation of the aquifer were set to 10 m and -10 m. The model parameter $0 < a < 1$ defines the start- and endpoint of a spline tracing the paleo-channel. The true solution is assumed to be $a = 0.15$, and the prior is defined as a beta distribution with parameters $\alpha, \beta = 2$. Observations are collected in three wells along the central north-south axis with an observation standard deviation of $\sigma = 0.025$ m. The model is implemented in MODFLOW 6 (Langevin et al., 2017) using the Python interface FloPy (Bakker et al., 2016).

We would like to draw attention to the fact that the setup of this scenario is symmetric with respect to the central north-south axis. As such, we would expect that there are two functionally indistinguishable solutions to the inference problem: $a = 0.15$ and $a = 0.85$. We test the algorithm with an ensemble of $N = 100$ particles, 100 iterations, an initial step-size of $\varepsilon_{i,0} = 10^{-4}$, an acceleration rate of $\alpha = 1.5$, and a similarity cutoff of $\beta = 0.75$. The kernel bandwidth was set to the mean distance to the $k =$ 25th nearest neighbor during each iteration.

4.5.2 Results

Results of the inference process are illustrated in Figure 4-5. The posterior parameter field (Figure 4-5a, b) reveals that the expect bimodal uncertainty structure was successfully recovered by the algorithm: half the ensemble places the channel at $a = 0.15$, the other half at $a = 0.85$. If this were a real scenario, this ambiguity could be resolved with additional geological information, or a new observation well located to the left or right of the mirror axis.

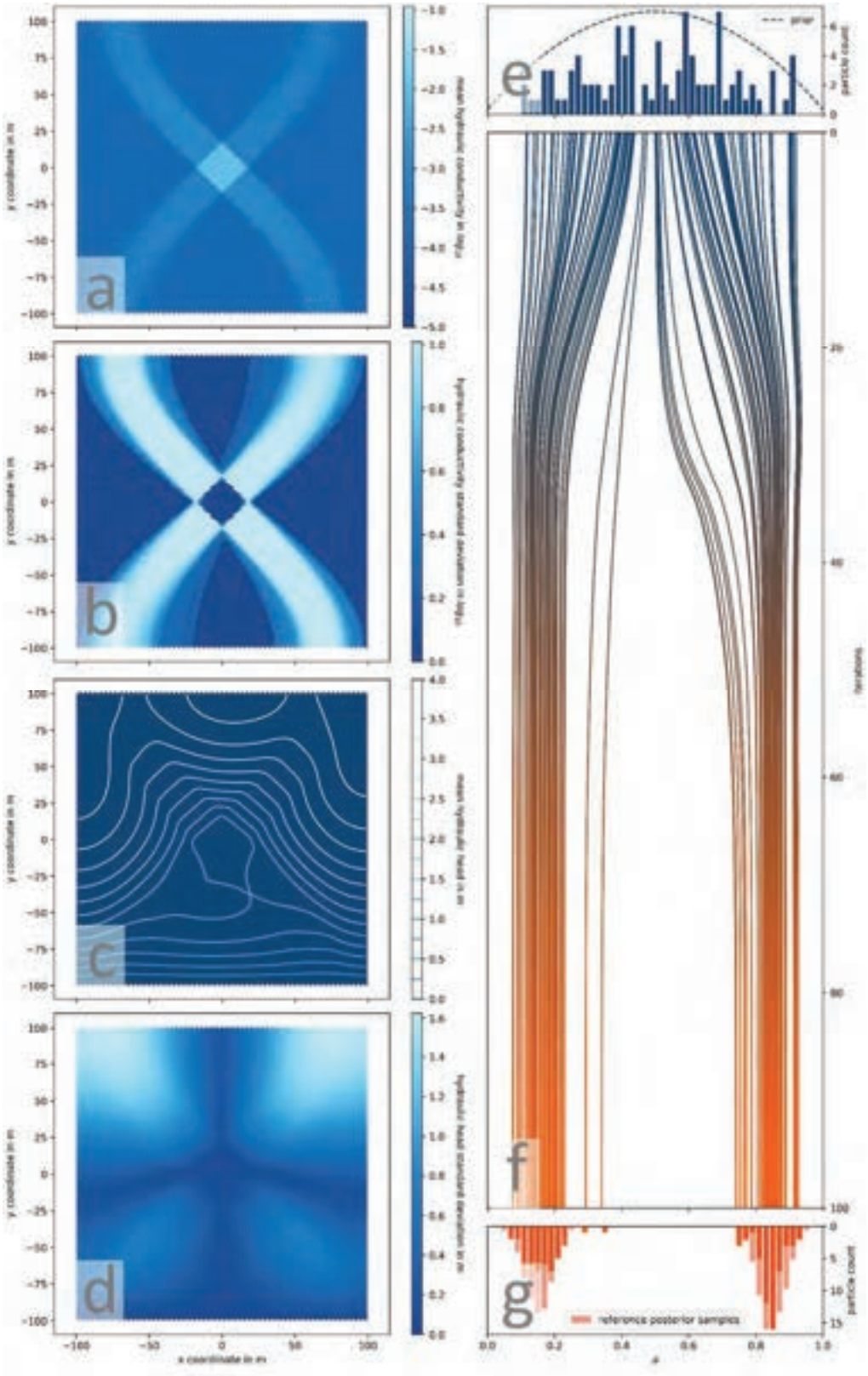


Figure 4-5. Results for the SVGD algorithm applied to the simple model: the left column shows the mean and standard deviation of hydraulic conductivity (a, b) and simulated head (c, d) at the end of the inference process. The right column illustrates the prior ensemble (e), the particle trajectories through the iterative process (f), and the resulting posterior ensemble (g).

To test if the algorithm truly converged against the posterior, we compare the posterior ensemble against results obtained from a long emcee (Foreman-Mackey et al., 2013) chain (Figure 4-5g, background)[16]. Figure 4-5f verifies that SVGD not only identified the correct posterior location, but also its spread.

4.6 Real test case

4.6.1 Site description

For the real test case, we focus on the Kempt valley in Switzerland, a small pre-alpine catchment located about 10 km east of the city of Zurich. Within the valley lies the city of Fehraltorf, surrounded by pastures for cows and horses. The valley is characterized as follows:

- **Geology**: The aquifer layout is highly heterogeneous, shaped by alpine geology and postglacial sedimentology. Multiple electric resistivity tomography campaigns failed to delineate the aquifer bottom, and the prevailing gravelly sediments preclude direct push coring past a depth of approximately 7 m. Geological maps and indirect information suggest north-eastern and south-western plateaus or banks or impermeable material (Figure 4-6a).
- **Hydro(geo)logy**: The groundwater table is generally shallow, sometimes ponding during spring or after large precipitation events. Consequently, large swathes of the valley are artificially dewatered with tile drainages. The central Kempt stream is only perennial past the city of Fehraltorf, where it is sustained by a local wastewater treatment plant (WWTP), drainage channels, and multiple culverted

[16] Then emcee samples are composed of 100 walkers making 445 jumps each, burn-in removed.

creeks (Vögeli, 2018). Upstream of Fehraltorf, the creek is called Luppmen and controlled almost exclusively by groundwater. Temporally, the groundwater table in the catchment is highly variable, particularly during the simulated drought year of 2018.

- **Infrastructure**: Due to the shallow groundwater table, the canalization beneath Fehraltorf (Figure 4-6c) is partially submerged and substantial groundwater infiltration is known to occur. We further have the extraction rates for two municipal pumping stations near the southern edge of the city (Figure 4-6e). The agricultural estates in the catchment and an industrial greenhouse vegetable farm have concessions for ground- and river water extraction, but unfortunately no quantitative rates were available for either. Due to the small size of these farms, we neglected these potential groundwater sinks as probably irrelevant to the overall water budget.
- **Boundary conditions**: Located in a headwater catchment, we expect that the valley receives significant inflow from the surrounding hillslopes (Figure 4-6f). We did not explicitly simulate these hillslopes, instead roughly delineating six upslope catchments based on topographic information. These upslope catchments form the basis of conceptual models with uncertain extent and temporal dynamics which define the time-variable inflow into the central model. Vertical recharge is applied without delay (due to the shallow nature of the aquifer) and estimated from the difference between precipitation measurements within Fehraltorf and spatially averaged measurements of actual evapotranspiration in surrounding stations.

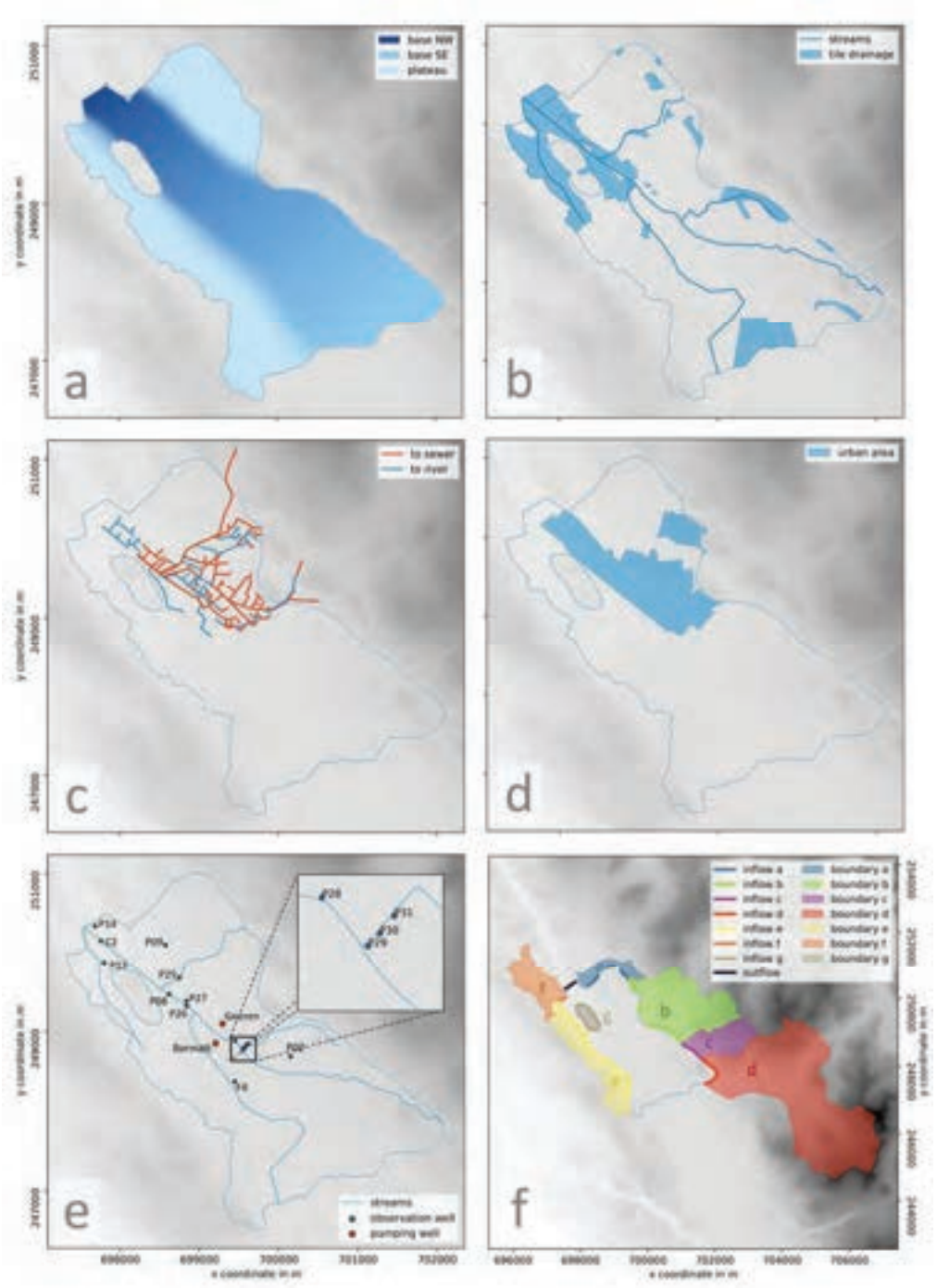


Figure 4-6. Approximate aquifer topology (a), tile drainage, open and culverted streams (b), extent of canalization (c) and urban area (d), location of pumping and observation wells (e), and upslope contributing areas (f).

We simulate the drought year of 2018 (January 1st to December 31st), during which groundwater extraction and use had to be restricted due to an exceedingly low water table. We initialize the model with a seven-month spin-up period starting June 1st 2017. We assimilate hydraulic head data from a number of observation wells (Figure 4-6e) as well as estimates of canalization groundwater infiltration rates obtained by component separation from an independent canalization model.

4.6.2 Model setup

We implement the numerical model in MODFLOW 6 (MF6: Langevin et al., 2017) using FloPy (Bakker et al., 2016). This framework permits a Newton-Raphson formulation for unstructured grids, which is more resilient to the drying of model cells. Furthermore, its modular structure and mover package permit the representation of the complex interactions of the stream, the canalization, the drainage system, and the groundwater. Capitalized three-letter acronyms in the following paragraph refer to the respective MF6 packages.

We tessellated the model domain with a single layer of 4079 hexagonal prisms. The depth of the aquifer bottom is defined by four parameters which specify the elevation of four masks: the northwest-to-southeast gradient, the north-eastern plateau, and the south-western plateau (Figure 4-6a). Hydraulic conductivity is extrapolated through inverse distance weighting (Shepard, 1968) from 30 nodes. Tile drainages, the culverted creeks, and the canalization are implemented as drainages (DRN), whose flows are diverted to their respective outflow points in the Luppmen through the mover (MVR) package. The conductance of the canalization elements is extrapolated from ten nodes, and implemented as what we shall call a pre-conductance, to be multiplied by the

canalization elements' surface area in order to yield the full conductance. Where streams (Figure 4-6b) are open, their bed elevation has been measured, where they are culverted, their elevation has been extrapolated. The tile drainages were assumed to be located 0.75 m below the surface. The two non-culverted streams, Luppmen (main stream from SE to NW, Figure 4-6b, f) and Wildbach (northernmost stream, from NE into Luppmen, Figure 4-6b, f) are represented with the surface flow routing (SFR) module, which permits exchange with groundwater in both directions. The riverbeds' hydraulic conductivity was set to -5 $\log_{10}$ m/s, the riverbed thickness to 30 cm, and its width to 3 m (Luppmen) or 1.5 m (Wildbach). Their Manning's coefficients are adaptable parameters. Direct runoff due to surface sealing in the urban areas is represented through a 35% flat recharge reduction. Infiltration into the canalization is separated into two parts: infiltration into storm drainages, and infiltration into the sewer system. The former is routed directly into the Luppmen. The latter is used for inference against an estimated groundwater fraction of the total wastewater treatment plant (WWTP) inflow. The total WWTP outflow – simulated groundwater infiltration plus domestic wastewater component – is routed into the Luppmen.

Recharge is estimated from the difference of average precipitation measurements around Fehraltorf and a spatially averaged evaporation estimate from Meteoswiss (2020). Since the groundwater table is shallow and the time steps – set to three hours each – are coarse, we assumed instantaneous recharge within the valley. Recharge on the hillslopes is routed into the valley through time-variable inflow boundaries (Figure 4-6f) according to a simple, conceptual forcing model (Figure 4-7). This

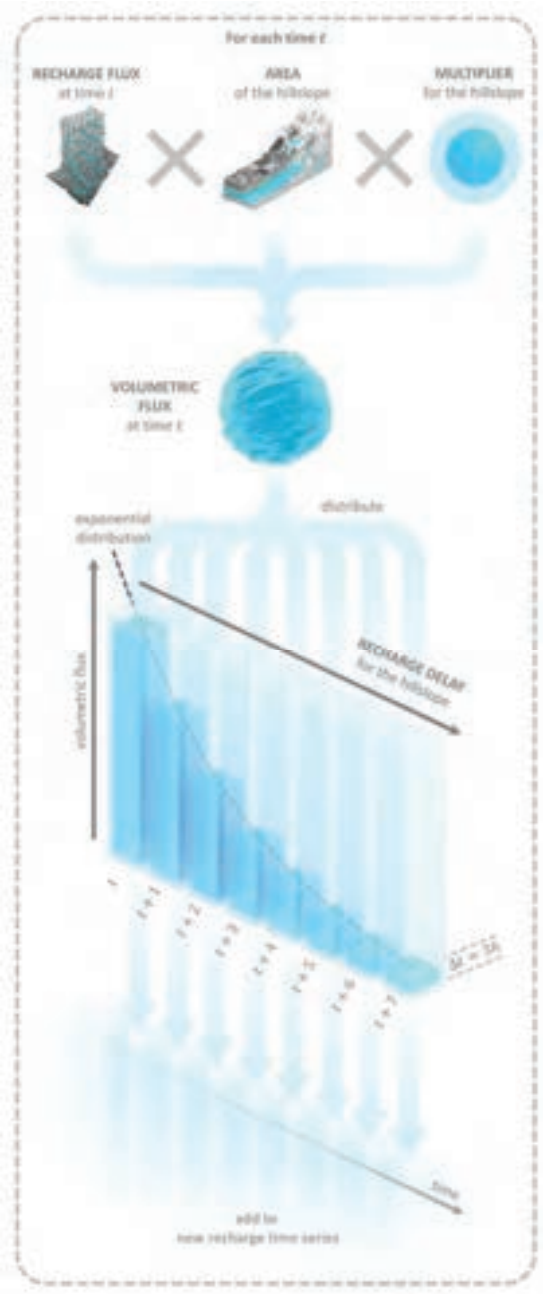


Figure 4-7. Illustration of the forcing model. For each time step and boundary, recharge estimates are multiplied by its area and a multiplier. The resulting volumetric flux is then distributed to subsequent timesteps according to an exponential distribution scaled by a recharge delay parameter. Finally, the distributed fluxes of each time are added up to yield the volumetric boundary inflow, distributed across its inflow model cells.

forcing model multiplies each timestep's raw recharge estimate with each boundary's upslope area (delineated based on topography) and a recharge multiplier. The latter is intended to compensate for potential deviations of the unknown groundwater catchment from the topographic catchment, bias in the recharge estimate, as well as unknown sinks or sources along the hillslopes. The resulting volumetric flux is then distributed among the subsequent timesteps according to an exponential distribution, whose extent is defined by a second parameter, the recharge delay. This parameter is intended to represent unresolved surface- and groundwater flow processes along the hillslope and controls the flashiness of the inflow. The temporally distributed volumetric flux components are then added to a new volumetric flux

time series, and the process is repeated for the next time step. Once the new time series is assembled, the volumetric fluxes are distributed spatially across the respective boundary's inflow cells (Figure 4-6f).

In total, the numerical model features $D = 61$ parameters, some of which (hydraulic conductivity nodes, aquifer bottom elevation nodes, and forcing model parameters) are first converted into grid parameters using deterministic pre-processors. The priors of the parameters are illustrated in Table 4-1.

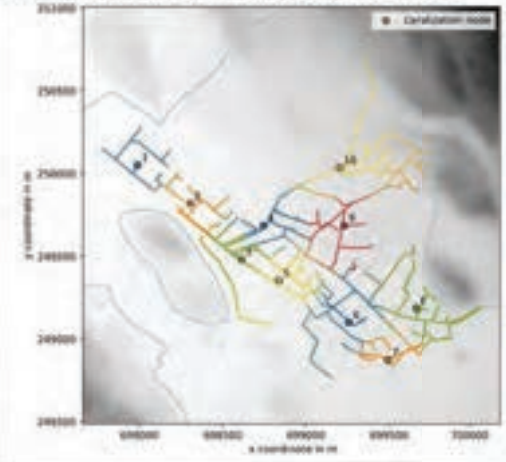
4.6.3 Algorithmic setup

We test the SVGD algorithm with two different ensemble sizes: an ensemble size of $N = 30$ and an ensemble size of $N = 100$. Considering the parameter space dimensionality of $D = 61$, the former scenario is restricted to exploring a subspace, while the latter scenario should have access to full parameter space. Consequently, we will focus on the $N = 100$ in the discussion of the results, as this scenario avoids the risk of misinterpreting optimization results. In both scenarios, we iterated 100 times. The required simulation time was about 30 hours for the $N = 30$ scenario, and about 102 hours for the $N = 100$ scenario.

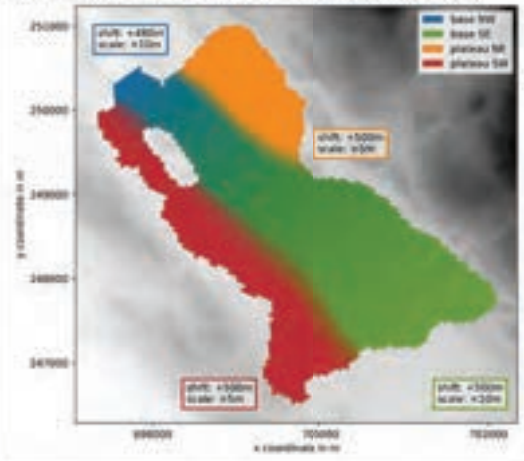
Table 4-1. Model parameters, priors and limits. Capitalized letters in the note column correspond to boundaries, LP refers to Luppmen, WB refers to Wildbach. Colored regions in map 1 and map 3 illustrate influence areas of different nodes. The ring above the north-eastern plateau in map 2 marks the mean of its slope orientation.

parameter		prior			limits		transformation	
name	note	pdf type	μ or α	σ or β	min	max	shift	scale
recharge delay	A-F	beta	5	7	+0.01	+0.99	+0	× 3
recharge multiplier	A-F	normal	0	1	−5	+5	+0	× 0.05
river flow fraction	B, D	beta	3	3	+0.01	+0.99	+0	× 1
Manning's coefficient	LP/WB	beta	3	5	+0.01	+0.99	+0.01	× 0.08
canalization pre-conductance	map 1	beta	2	5	+0.01	+0.99	−7	× 7
aquifer bottom elevation	map 2	normal	0	1	None	None	see map 2	see map 2
hydraulic conductivity	map 3	normal	0	1	−3	+3	−4	× 0.5
specific yield	None	beta	5	15	+0	+1	+0	× 0.5

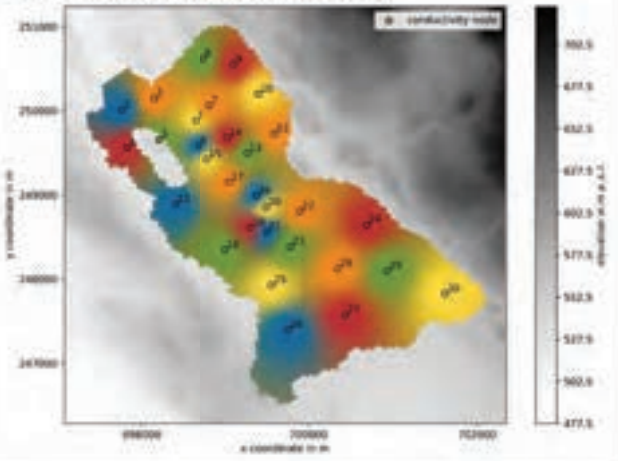
map 1: *interpolation of canalization pre-conductance*



map 2: *interpolation and shift/scale of aquifer topology*



map 3: *interpolation of hydraulic conductivity*



4.6.4 Results

The simulated states at the observation wells and the canalization for the posterior ensemble of the $N = 100$ scenario are illustrated in Figure 4-8, for the prior ensemble and the scenario $N = 30$ in Figure III-1 and Figure III-2 (supporting information). Improvements to the simulated hydraulic heads are significant, reducing the root mean square error (RMSE) from a prior average of 312 cm down to a posterior average of 30 cm (Figure 4-9) for the scenario $N = 100$, and from 322 cm down to 39 cm for the scenario $N = 30$ (Figure III-2). Proportionally, bias is reduced even further, from a prior mean of 207 cm down to a posterior mean of only 4 cm in the case of $N = 100$, and from 201 cm to 2 cm for $N = 30$. The slightly elevated RMSE contrasted by very low bias suggests that the residual error is rooted in model structural deficiencies.

We expect a significant impact from such model deficiencies since we only employed a single prescribed head boundary at the outflow. Consequently, all hydraulic head fluctuations within the model domain must be created by the model itself, instead of being partially inherited from a hypothetical upslope prescribed head boundary. The simulated and observed hydraulic heads seem to support this interpretation (Figure 4-8). The model successfully recreated the yearly dynamics in most wells, but we can observe varying patterns between them, often with errors which may have a plausible model-structural explanation.

Wells C2 and P08 (Figure 4-8a and d), for example, barely fluctuate over the year and retain a relatively steady water level. This suggests that both wells are subjected to some form of stabilizing influence, likely a perennial drainage effect of some sort. Both wells are located adjacent to the Luppmen and the canalization, but only the latter qualifies as a

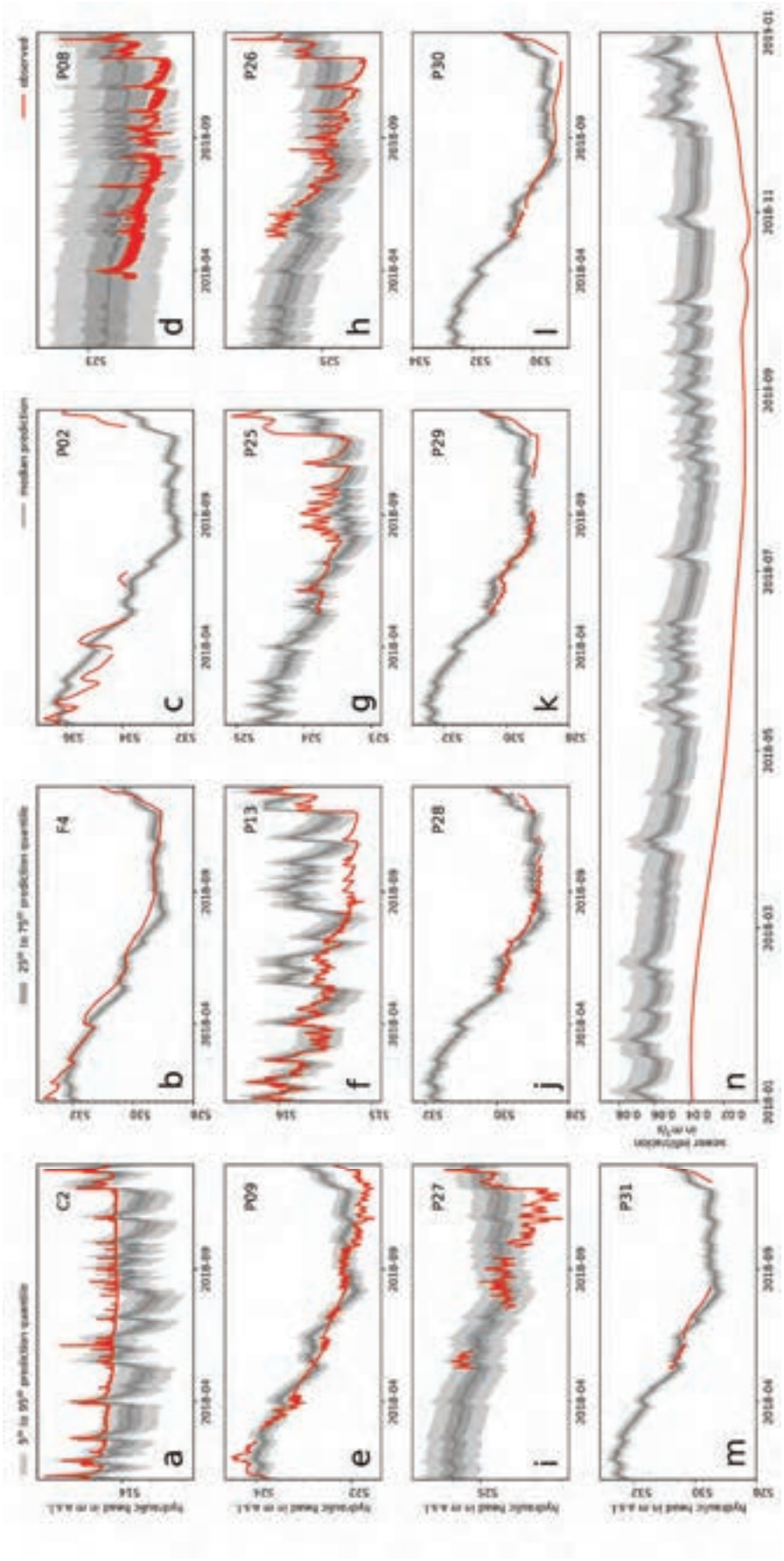


Figure 4-8. Posterior simulated (greyscale) and observed (red) hydraulic heads (a-m) and canalization groundwater infiltration (n) with model error at the end of simulation period for $N = 100$. Prior results are illustrated in Figure III-3.

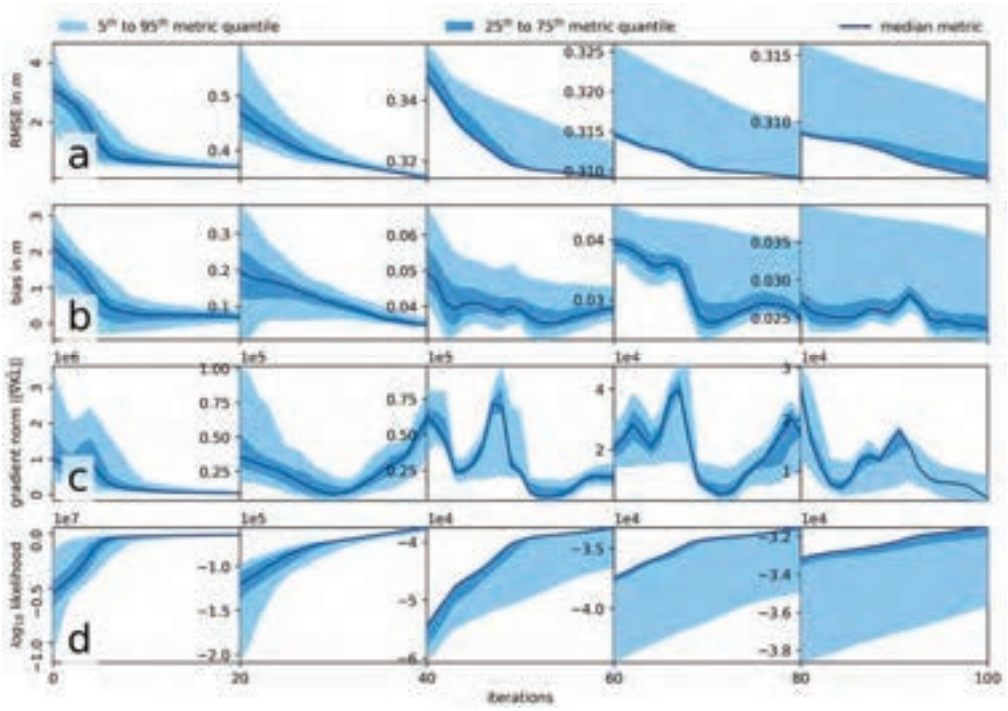


Figure 4-9. Posterior overall root-mean square error (o) and bias (p) for the hydraulic heads, the mean norm of the Kullback-Leibler divergence gradient (q) and the log-likelihood (r) across the algorithm's iterations. For better visualization, the y-axis scale is reset every 20 iterations for the scenario $N = 100$.

potential drainage, since the riverbed is above the water table at both locations. As can be seen in the parameter results, the model increased the canalization's pre-conductance near both wells (Figure 4-10g), successfully suppressing the yearly fluctuations. However, due to our simplified representation of the canalization and its leakage, both water tables are stabilized at somewhat wrong levels. To improve model fidelity, a finer resolution of the canalization breaches might be required. This hypothesis is also supported by the groundwater infiltration into the canalization (Figure 4-9n), which is consistently overestimated, thus suggesting that infiltration might occur more point-wise.

Wells F4, P02, P09, and P28 to P31 (Figure 4-8b, c, e, j-m) feature similar yearly trends, recovered to varying degrees of fidelity: A steady water table decrease by up to 3 m from January to September, followed by a

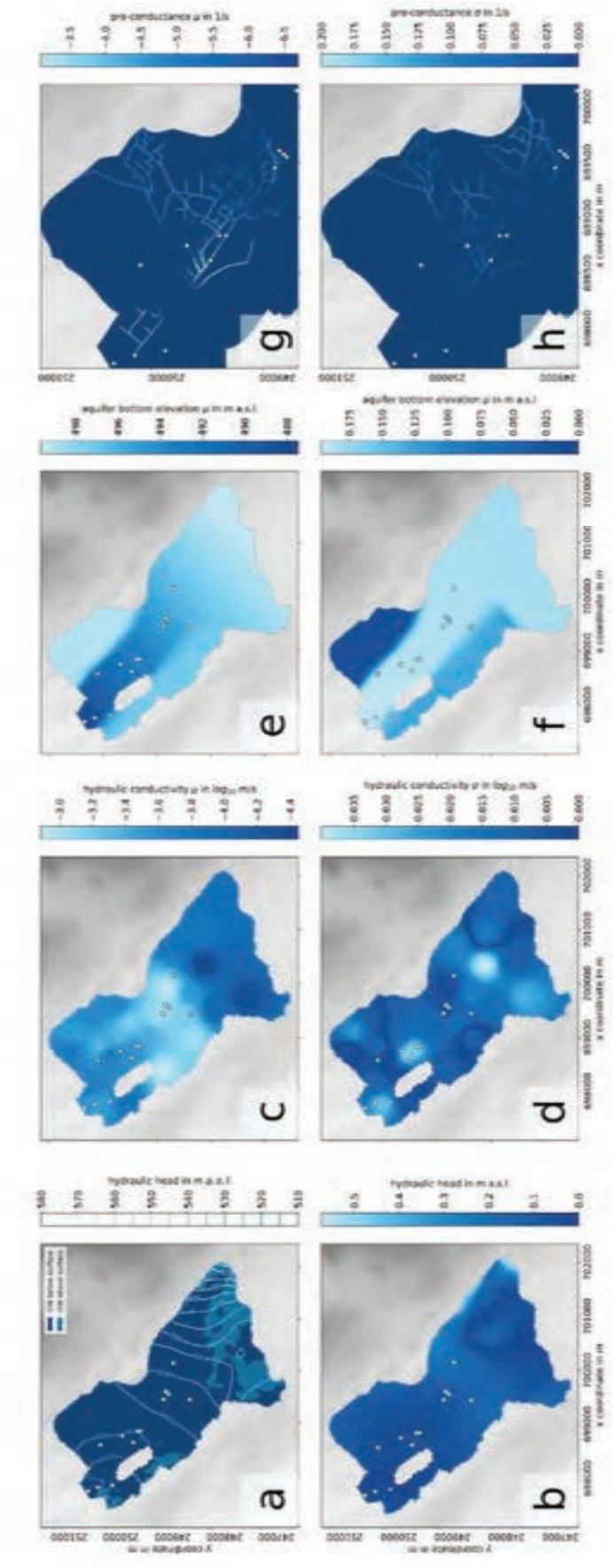


Figure 4-10. Posterior parameters and hydraulic head at the final iteration for $N = 100$. The two rows illustrate mean (a, c, e, g) and standard deviations (b, d, f, h) of hydraulic head in the initial steady-state simulation period (c, d), hydraulic conductivity (e, f), aquifer bottom elevation (g, h), and canalization conductance (i, j). Corresponding prior fields are illustrated in Figure III-7.

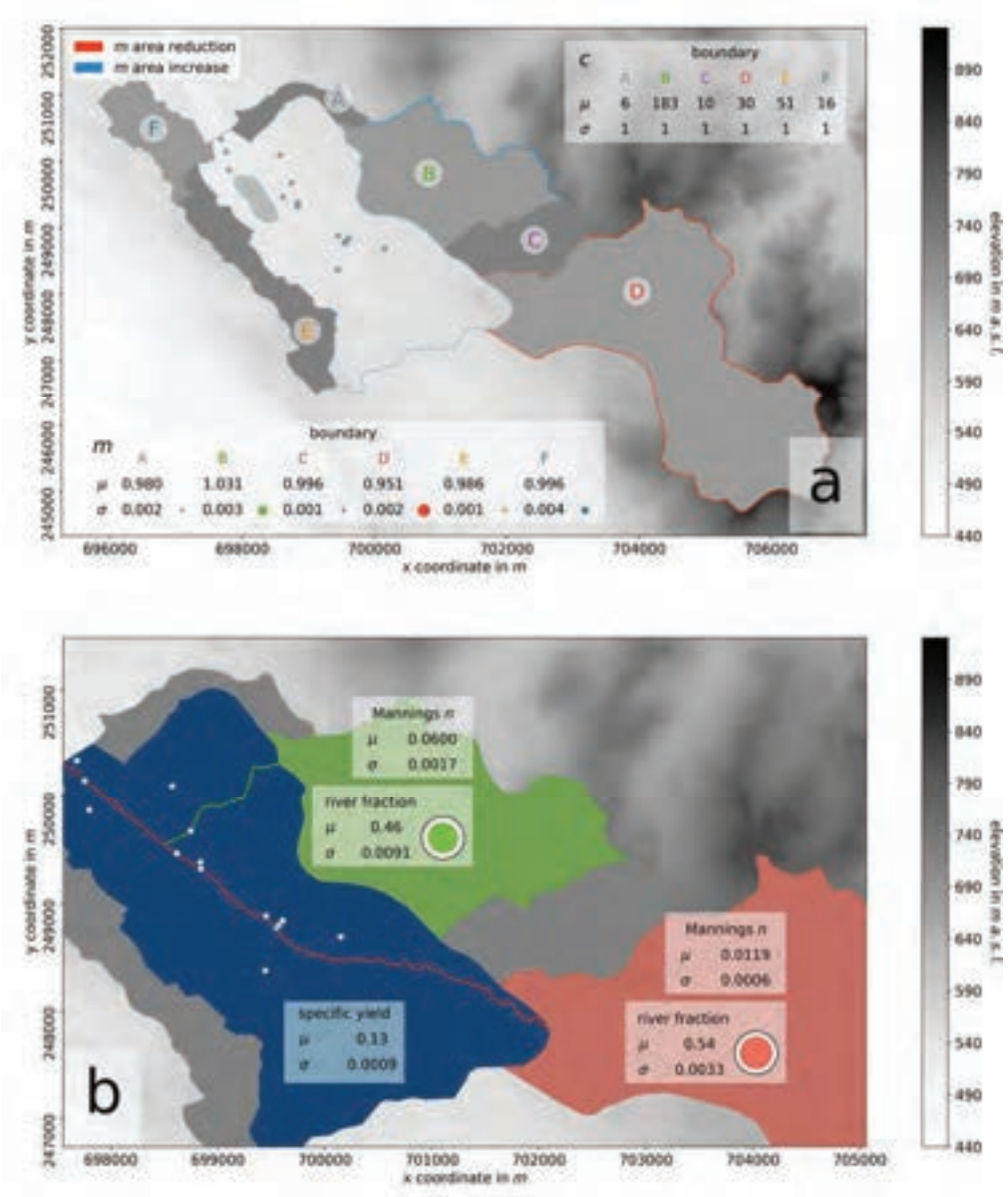


Figure 4-11. Posterior values for specific yield and the forcing-related parameters at the final iteration for $N = 100$. Subplot (a) illustrates the boundary-related parameters: the scalar multiplier m and the recharge delay factor d in hours; mean m is illustrated as a faint outline around the boundaries, visualizing the area inflation or deflation. Subplot (b) shows the specific yield and river-related parameters: the river discharge fraction and the Manning's number for Wildbach and Luppmen, respectively. Corresponding prior fields are illustrated in Figure III-8.

steep rebound in late autumn. While the water table drop is reproduced faithfully, its rebound is underestimated in all wells. This suggests that the aquifer's storage capacity may have been overestimated. Possible causes of such an overestimation could be missing, perennial source terms. Potential candidates could be leakage from water mains, or vertical inflow from hypothetical deeper, semi-confined aquifers.

The patterns in the remaining wells are somewhere between the two sets discussed above. P13 (Figure 4-8f) is located in an agricultural area with tile drainages and diverges from the observed water tables only from May onwards. The remaining wells (P25-P27, Figure 4-8g-i) are located in the urban area of Fehraltorf and feature fluctuations which the model cannot seem to fully recreate.

Overall, it seems our prior parameter assumptions resulted in an initial overestimation of water tables, which SVGD corrected by reducing hydraulic conductivities (Figure 4-11c) relative to the prior, particularly near the centre of the catchment. Individual changes to parameter uncertainty for the scenarios $N = 100$ and $N = 30$ are illustrated in Figure III-11 and Figure III-12 (supporting information). The model simulates groundwater ponding in the initial steady-state spin-up period near the southern and western edges of the valley (Figure 4-11a). This ponding may not be unrealistic, as both areas feature tile drainages and very shallow water tables. Particularly in the southern region we have some evidence for ponding: a naturally marshy, extensively drained forest, and a small airfield whose runway is often closed during spring due to flooding.

4.7 Discussion

In summary, the inference results of SVGD were promising, returning the true posterior in the synthetic test case, and yielding substantial improvements in terms of predictive error for the real test case. In the latter scenario, the observed states did not always remain within the error bounds, which suggests both structural model inadequacy and an underestimation of the model error. We identified some potential

sources of this error – the omission of agricultural irrigation, and low-resolution canalization drainage –, which could be revised in a future iteration of the conceptual model. The standard deviation of the model error is a parameter which could also be inferred, although we note that this would complicate the derivative of the loglikelihood gradient (Equation 4-17). A further interesting addition would be the consideration of temporal correlation in the model error covariance matrix, which may prevent the strong tapering of the posterior in the real test case.

As far as the inference itself is concerned, SVGD successfully recovered the synthetic bimodal posterior – a nigh-impossible task for non-localized methods based on the assumption of Gaussianity. In the real test case, results for the subspace-limited $N = 30$ scenario were also promising. While we acknowledge that it is undesirable and potentially dangerous to restrict parameter inference to a subspace, computational limitations often necessitate working with such restrictions. This ability to recover at least simplified uncertainty estimates in settings with inevitably insufficient computational resources constitutes, in our opinion, one of the main advantages of the EnKF, and is shared by our implementation of SVGD.

Despite the promising optimization performance, this algorithm comes at a computational price: the necessity to iterate requires re-simulating the entire observation history for each particle during every iteration, whereas filter methods like the EnKF must only simulate the model history once for each particle (separated into distinct assimilation time steps). However, it may not be necessary to iterate for as long as we did in our test cases – towards the end, improvements were only minor. We

remain confident that performance can be improved further with adjustments to the gradient descent algorithm.

4.8 Conclusions

In this study, we employed the Stein Variational Gradient Descent algorithm of Liu & Wang (2016) and proposed adaptations necessary for its practical application for non-Gaussian parameter inference in hydrogeological models. Towards this end, we proposed a computationally inexpensive, localized, ensemble-based approximation of the Jacobian. This matrix is a prerequisite for the estimation of the logposterior gradient, and possibly the greatest computational obstacle to the implementation of SVGD. We also proposed a simple gradient descent algorithm which optimizes the algorithm's computational efficiency by adapting the descent step size dynamically.

We then proceeded to illustrate the performance of the algorithm in two test cases: a simple synthetic model with an intuitive solution, and a complex model based on a real field site with non-trivial, nonlinear parameter interactions. Results in both cases were promising. Our application in the synthetic test case successfully converged against the bimodal reference solution obtained by MCMC, iteratively evolving a unimodal prior into a bimodal posterior – an impossible task for Gaussian algorithms such as the EnKF. While no reference solution was available for the real test case, inference results seemed promising as well. Despite the model's complexity, the inference significantly reduced simulation error and bias, with the residual error likely being based on model structural errors. Throughout, the algorithm retained uncertainty without the need for artificial variance inflation, a challenge for particle

filters (e.g., Ramgraber et al., 2019, 2020) or the EnKF (Anderson, 2007). We tested the algorithm's performance (and the fidelity of our approximations) for two ensemble sizes: $N = 30$, restricted to an at most 29-dimensional parameter-subspace, and $N = 100$, with theoretical access to all parameter space dimensions. Substantial improvements were obtained in both scenarios, although the larger ensemble size yielded slightly better optimization results.

A limitation of this algorithm is its restriction to smooth probability distributions with at least convex support, a weakness shared with other gradient descent algorithms. For the inference of structural uncertainty of geological facies, it may be necessary to employ an auxiliary parameterization which permits a smooth pdf first (e.g., Hu et al., 2013; Ramgraber et al., 2019). A further possible source of error may be found in our ensemble-based Jacobian approximation. While our synthetic example converged successfully and optimization results were promising in both test cases, we cannot guarantee that this approximation proves adequate in all cases.

For future research, we are optimistic that the experimentation with other gradient descent algorithms could improve the efficiency of the SVGD algorithm even further. Alternative Jacobian approximations, particularly those obtained with automatic differentiation, seem a promising way to improve the fidelity of practical applications of SVGD and constitute an important avenue for future research. Other fascinating research directions could be found in the related field of transport maps (e.g., Marzouk et al., 2017; El Moselhy & Marzouk, 2012; Spantini et al., 2018) which construct the transformation functions explicitly. In conclusion, we believe that SVGD is a highly promising and

relatively easy-to-use (although not necessarily easy-to-derive) tool for non-Gaussian parameter inference in hydrogeological systems, and that a strong case could be made for its use in complex models with questionable claim to Gaussianity.

4.9 Acknowledgements

We express our gratitude to Dr. Carlo Albert, Swiss Federal Institute of Aquatic Science and Technology (Eawag), for many discussions and guidance during the interpretation and re-derivation of the SVGD algorithm. We furthermore extend our gratitude to Prof. Manuel Pulido, University of Reading, for providing the source code of his publication's examples, which aided the troubleshooting of our own algorithm. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675120.

4.10 Appendix 1: Kernelized Stein Discrepancy

Kernelized Stein Discrepancy (KSD) is based on Stein's identity (Stein, 1972; Stein et al., 2004), which states that for certain vector-valued functions $\boldsymbol{\phi}$ (functions in the *Stein Class* of p), we have:

$$\mathbb{E}_{\boldsymbol{\theta}\sim p}\left[\boldsymbol{A}_p\boldsymbol{\phi}(\boldsymbol{\theta})\right] = 0 \qquad 4\text{-}38$$

where $\mathbb{E}_{\boldsymbol{\theta}\sim p}$ denotes the expectation under the assumption that $\boldsymbol{\theta}$ is sampled from a distribution p (note that in the following, p will refer to the target distribution, i.e. the posterior), and $\boldsymbol{A}_p$ is a linear operator (the *Stein operator*) on some function $\boldsymbol{\phi}$:

$$\boldsymbol{A}_p\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{\phi}(\boldsymbol{\theta})[\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta})]^\top + \boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{\phi}(\boldsymbol{\theta}) \qquad 4\text{-}39$$

Assuming that $\boldsymbol{\phi}(\boldsymbol{\theta})$ is a D-dimensional vector

$$\boldsymbol{\phi}(\boldsymbol{\theta}) = [\phi_1(\boldsymbol{\theta}), \dots, \phi_D(\boldsymbol{\theta})]^\top \qquad 4\text{-}40$$

and $\boldsymbol{\theta} \in \mathbb{R}^D$, $\boldsymbol{A}_p\boldsymbol{\phi}(\boldsymbol{\theta})$ will be a $D \times D$-matrix. Liu et al. (2016) note that $\boldsymbol{\phi}$ does not have to be a D-valued vector function and can even be scalar, but for the purpose of this derivation we assume its output matches the dimensionality of parameter space $\mathbb{R}^D$. What Equation 4-38 effectively states is that for functions $\boldsymbol{\phi}$ in a certain class $\mathcal{F}$ ($\boldsymbol{\phi} \in \mathcal{F}$), the expectation of $\boldsymbol{A}_p\boldsymbol{\phi}(\boldsymbol{\theta})$ will be zero if $\boldsymbol{\theta}\sim p$. This is nice, but not particularly useful on its own.

A more interesting case occurs when we are not certain whether the samples $\boldsymbol{\theta}$ are drawn from the distribution p. If we assume that they might instead be samples of a second distribution q, the expectation in Equation 4-38 will no longer generally equal zero:

$$\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\boldsymbol{A}_p\boldsymbol{\phi}(\boldsymbol{\theta})\right] \neq 0 \quad if\ q \neq p \qquad 4\text{-}41$$

This expression can be used to obtain a discrepancy measure for two distributions, one which does not depend on the normalization factor (or *model evidence*) $p(\boldsymbol{y})$ of Bayes' theorem. Unfortunately, Equation 4-41 no longer yields the same result for all $\boldsymbol{\phi} \in \mathcal{F}$. As such, we would wish to find the most discriminant function $\boldsymbol{\phi}^* \in \mathcal{F}$ which maximizes Equation 4-41:

$$\mathbb{S}(q||p) = \max_{\boldsymbol{\phi}\in\mathcal{F}} \left\{ \left[\mathbb{E}_{\boldsymbol{\theta}\sim q} \left[\text{trace}\, \boldsymbol{A}_p \boldsymbol{\phi}(\boldsymbol{\theta}) \right] \right]^2 \right\} = \left[\mathbb{E}_{\boldsymbol{\theta}\sim q} \left[\text{trace}\, \boldsymbol{A}_p \boldsymbol{\phi}^*(\boldsymbol{\theta}) \right] \right]^2 \qquad 4\text{-}42$$

where we have taken the trace of the matrix to obtain a scalar value, and the expectation has been squared to render the optimization objective irrespective of sign. Liu et al. (2016) suggest that matrix norms other than the trace might also be possible.

4.11 Appendix 2: Functional optimization in KSD

We want to find the vector field $\boldsymbol{\phi}^*$ which maximizes the KSD $\mathbb{S}(q||p)$.

$$\mathbb{S}(q||p) = \max_{\boldsymbol{\phi}\in\mathcal{F}} \left\{ \left[\mathbb{E}_{\boldsymbol{\theta}\sim q} [\text{trace}\, \boldsymbol{A}_p \boldsymbol{\phi}(\boldsymbol{\theta})] \right]^2 \right\} = \left[\mathbb{E}_{\boldsymbol{\theta}\sim q} [\text{trace}\, \boldsymbol{A}_p \boldsymbol{\phi}^*(\boldsymbol{\theta})] \right]^2 \quad 4\text{-}43$$

If we use the trace identity (the trace of an outer product is the inner product) where $\boldsymbol{a}$ and $\boldsymbol{b}$ are D-dimensional column vectors

$$\text{trace}\, \boldsymbol{a}\boldsymbol{b}^\top = \boldsymbol{a}^\top \boldsymbol{b} = \sum_{d=1}^{D} a_d b_d \quad 4\text{-}44$$

and Equation 4-11 in Equation 4-43, we obtain:

$$\mathbb{S}(q||p) = \left[\mathbb{E}_{\boldsymbol{\theta}\sim q} \left[\sum_{d=1}^{D} \phi_d^*(\boldsymbol{\theta}) \frac{\partial}{\partial \theta_d} \log p(\boldsymbol{\theta}) + \frac{\partial}{\partial \theta_d} \phi_d^*(\boldsymbol{\theta}) \right] \right]^2 \quad 4\text{-}45$$

where $\phi_d^*(\boldsymbol{\theta})$ is the d-th entry of $\boldsymbol{\phi}^*(\boldsymbol{\theta})$, and $\partial/\partial\theta_d$ is the partial derivative at $\boldsymbol{\theta}$ in parameter space dimension d. We can further reformulate Equation 4-45 by taking the square root of both sides and pulling the expectation operator inside the sum:

$$\sqrt{\mathbb{S}(q||p)} = \sum_{d=1}^{D} \mathbb{E}_{\boldsymbol{\theta}\sim q} \left[\phi_d^*(\boldsymbol{\theta}) \frac{\partial}{\partial \theta_d} \log p(\boldsymbol{\theta}) + \frac{\partial}{\partial \theta_d} \phi_d^*(\boldsymbol{\theta}) \right] \quad 4\text{-}46$$

Unfortunately, Equation 4-46 alone does not help us very much since we still do not know the function $\boldsymbol{\phi}^*$. To solve the optimization in Equation 4-43, we assume that the vector elements of $\boldsymbol{\phi}(\boldsymbol{\theta})$ are scalar-valued functions $\phi_d(\boldsymbol{\theta}), d = 1, \dots, D$

$$\boldsymbol{\phi}(\boldsymbol{\theta}) = [\phi_1(\boldsymbol{\theta}), \dots, \phi_D(\boldsymbol{\theta})]^\top \quad 4\text{-}47$$

where each $\phi_d(\boldsymbol{\theta})$ is defined in a RKHS $\mathcal{H}$. If we use the reproducing property (Equation 4-9) in Equation 4-46, we have:

$$\sqrt{\mathbb{S}(q||p)} = \sum_{d=1}^{D} \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\langle \boldsymbol{\varphi}_d^*(\cdot), \boldsymbol{k}(\cdot,\boldsymbol{\theta})\rangle_{\mathcal{H}} \frac{\partial}{\partial \theta_d} \log p(\boldsymbol{\theta}) + \frac{\partial}{\partial \theta_d}\langle \boldsymbol{\varphi}_d^*(\cdot), \boldsymbol{k}(\cdot,\boldsymbol{\theta})\rangle_{\mathcal{H}}\right] \quad 4\text{-}48$$

where we define $\boldsymbol{\varphi}_d^*(\cdot) = \left[\varphi_1^*/\sqrt{\lambda_1}, \dots, \varphi_\infty^*/\sqrt{\lambda_\infty}\right]^T$ as a vector in $\mathcal{H}$ which uniquely defines ϕ_d^* through $\phi_d^*(\boldsymbol{\theta}) = \langle \boldsymbol{\varphi}_d^*(\cdot), \boldsymbol{k}(\cdot,\boldsymbol{\theta})\rangle_{\mathcal{H}}$. Since $\frac{\partial}{\partial \theta_d}\log p(\boldsymbol{\theta})$ and $\frac{\partial}{\partial \theta_d}$ are scalar, we can pull them inside the inner products:

$$\sqrt{\mathbb{S}(q||p)} = \sum_{d=1}^{D} \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\langle \boldsymbol{\varphi}_d^*(\cdot), \frac{\partial}{\partial \theta_d}\log p(\boldsymbol{\theta})\, \boldsymbol{k}(\cdot,\boldsymbol{\theta})\rangle_{\mathcal{H}} + \langle \boldsymbol{\varphi}_d^*(\cdot), \frac{\partial}{\partial \theta_d}\boldsymbol{k}(\cdot,\boldsymbol{\theta})\rangle_{\mathcal{H}}\right] \quad 4\text{-}49$$

Now we can combine the two inner products and pull the expectation into the inner product:

$$\sqrt{\mathbb{S}(q||p)} = \sum_{d=1}^{D} \langle \boldsymbol{\varphi}_d^*(\cdot), \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\frac{\partial}{\partial \theta_d}\log p(\boldsymbol{\theta})\, \boldsymbol{k}(\cdot,\boldsymbol{\theta}) + \frac{\partial}{\partial \theta_d}\boldsymbol{k}(\cdot,\boldsymbol{\theta})\right]\rangle_{\mathcal{H}} \quad 4\text{-}50$$

To use this result, let us reflect on some of its properties: First, consider the case $D = 1$ for simplicity. In this case, the sum in Equation 4-50 disappears, and $\sqrt{\mathbb{S}(q||p)}$ is maximized when the inner product is maximized. From the geometric interpretation of an inner product, we know that it is maximized when both vectors are

a) collinear, and
b) pointing in the same direction[17].

This is already useful, but to obtain a unique solution we do not only need the direction of the vector $\boldsymbol{\varphi}_d^*(\cdot)$, but also its length (i.e., its *norm*). As such, we could somewhat arbitrarily restrict the norm to unity

[17] Recall the geometric interpretation of the inner product: An inner product between two vectors is (i) maximized if both vectors are collinear and pointing in the same direction, (ii) minimized if both vectors are collinear and pointing in opposite directions, and (iii) zero if both vectors are orthogonal.

($\|\boldsymbol{\varphi}_d^*(\cdot)\|_{\mathcal{H}} = 1$) and thus optimize $\boldsymbol{\varphi}_d^*(\cdot)$ on the unit ball of $\mathcal{H}$. In this case, we can define

$$\boldsymbol{\varphi}_d^*(\cdot) = \frac{\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\frac{\partial}{\partial\theta_d}\log p(\boldsymbol{\theta})\,\boldsymbol{k}(\cdot,\boldsymbol{\theta}) + \frac{\partial}{\partial\theta_d}\boldsymbol{k}(\cdot,\boldsymbol{\theta})\right]}{\left\|\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\frac{\partial}{\partial\theta_d}\log p(\boldsymbol{\theta})\,\boldsymbol{k}(\cdot,\boldsymbol{\theta}) + \frac{\partial}{\partial\theta_d}\boldsymbol{k}(\cdot,\boldsymbol{\theta})\right]\right\|_{\mathcal{H}}} \tag{4-51}$$

Still considering the scenario $D = 1$, we can simplify Equation 4-51 using the Stein operator:

$$\boldsymbol{\varphi}_d^*(\cdot) = \frac{\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[A_p\boldsymbol{k}(\cdot,\boldsymbol{\theta})\right]^{(d)}}{\left\|\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[A_p\boldsymbol{k}(\cdot,\boldsymbol{\theta})\right]\right\|_{\mathcal{H}}} \tag{4-52}$$

Since the vector $\boldsymbol{\varphi}_d^*(\cdot)$ defines the function ϕ_d^* in $\mathcal{H}$ (scalar because we assume $D = 1$), we can retrieve the corresponding function with the reproducing property (Equation 4-9):

$$\phi_d^*(\boldsymbol{\theta}') = \langle\boldsymbol{\varphi}_d^*(\cdot), \boldsymbol{k}(\cdot,\boldsymbol{\theta}')\rangle_{\mathcal{H}} \propto \langle\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[A_p\boldsymbol{k}(\cdot,\boldsymbol{\theta})\right], \boldsymbol{k}(\cdot,\boldsymbol{\theta}')\rangle_{\mathcal{H}} \tag{4-53}$$

where $\propto$ denotes proportionality. Since the KSD $\mathbb{S}(q||p)$, if used as a relative metric, does not really care about proportionality, we omit the proportionality from here on out. Continuing, inner product spaces like $\mathcal{H}$ have linearity in the first argument, so we can obtain:

$$\phi_d^*(\boldsymbol{\theta}') = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[A_p\langle\boldsymbol{k}(\cdot,\boldsymbol{\theta}), \boldsymbol{k}(\cdot,\boldsymbol{\theta}')\rangle_{\mathcal{H}}\right] \tag{4-54}$$

If we then use Equation 4-6, we obtain:

$$\phi_d^*(\boldsymbol{\theta}') = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[A_p k(\boldsymbol{\theta},\boldsymbol{\theta}')\right] \tag{4-55}$$

With this, we are almost finished. Let us now consider the case of $D > 1$. Note that the argument of the expectation in Equation 4-51, that is $\frac{\partial}{\partial\theta_d}\log p(\boldsymbol{\theta})\,\boldsymbol{k}(\cdot,\boldsymbol{\theta}) + \frac{\partial}{\partial\theta_d}\boldsymbol{k}(\cdot,\boldsymbol{\theta})$, can be interpreted as the d-th row of a $D \times \infty$ matrix $\boldsymbol{A}_p\boldsymbol{k}(\cdot,\boldsymbol{\theta})$ instead of the $1 \times \infty$ vector $A_p\boldsymbol{k}(\cdot,\boldsymbol{\theta})$ we had for $D = 1$. By extension:

$$\phi_d^*(\boldsymbol{\theta}') = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[A_p k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\boldsymbol{A}_p k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right]^{(d)} \qquad 4\text{-}56$$

where the superscript (d) refers to the row. In essence, the optimization we made above can be made for each parameter space dimension $d = 1, \dots, D$ individually (the only influence of the other dimensions arises from the normalization in Equation 4-52), which is equivalent to maximizing the inner product on a compositive Hilbert space $\mathcal{H}^D = \mathcal{H}_1 \times \dots \times \mathcal{H}_D$. If we combine these solutions, we obtain the result for a vector field $\boldsymbol{\phi}^*$ with D-dimensional output:

$$\begin{aligned}
\boldsymbol{\phi}^*(\boldsymbol{\theta}') &= \begin{bmatrix} \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\boldsymbol{A}_p k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right]^{(1)} \\ \vdots \\ \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\boldsymbol{A}_p k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right]^{(D)} \end{bmatrix} \\
&= \begin{bmatrix} \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[k(\boldsymbol{\theta}, \boldsymbol{\theta}')\frac{\partial}{\partial\theta_1}\log p(\boldsymbol{\theta}) + \frac{\partial}{\partial\theta_1}k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right] \\ \vdots \\ \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[k(\boldsymbol{\theta}, \boldsymbol{\theta}')\frac{\partial}{\partial\theta_d}\log p(\boldsymbol{\theta}) + \frac{\partial}{\partial\theta_d}k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right] \end{bmatrix} \\
&= \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[k(\boldsymbol{\theta}, \boldsymbol{\theta}')\begin{bmatrix} \frac{\partial}{\partial\theta_1} \\ \vdots \\ \frac{\partial}{\partial\theta_d} \end{bmatrix}\log p(\boldsymbol{\theta}) + \begin{bmatrix} \frac{\partial}{\partial\theta_1} \\ \vdots \\ \frac{\partial}{\partial\theta_d} \end{bmatrix}k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right] \\
&= \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\boldsymbol{A}_p k(\boldsymbol{\theta}, \boldsymbol{\theta}')\right] \\
&= \mathbb{E}_{\boldsymbol{\theta}\sim q}[k(\boldsymbol{\theta}, \boldsymbol{\theta}')\boldsymbol{\nabla}_{\boldsymbol{\theta}}\log p(\boldsymbol{\theta}) + \boldsymbol{\nabla}_{\boldsymbol{\theta}}k(\boldsymbol{\theta}, \boldsymbol{\theta}')]
\end{aligned} \qquad 4\text{-}57$$

This yields the desired solution.

4.12 Appendix 3: Relation to KLD

With the vector field $\boldsymbol{\phi}^*(\boldsymbol{\theta}')$ established, let us investigate how this function can be used for variational inference. In SVGD, our goal is to iteratively transform arbitrary particles into samples of the posterior. Towards this end, consider linear, invertible transformations of the form:

$$\boldsymbol{T}(\boldsymbol{\theta}) = \boldsymbol{\theta} + \varepsilon\boldsymbol{\phi}(\boldsymbol{\theta}) \tag{4-58}$$

where ε is a small scalar increment and $\boldsymbol{\phi}(\boldsymbol{\theta})$ is an evaluation of a vector field informing the descent direction. If we denote by q an arbitrary distribution after the transformation, and by $q_{[\boldsymbol{T}^{-1}]}$ the distribution after the transformation, we obtain by change of variables (see *Appendix 4: Change of variables*):

$$q_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = q\big(\boldsymbol{T}(\boldsymbol{\theta})\big) \cdot \left|\det\big(\nabla_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\big)\right| \tag{4-59}$$

In this appendix, we will outline the connection between KSD and KLD. KLD is an asymmetric discrepancy measure between two distributions q and p:

$$KL(q||p) = \int q(\boldsymbol{\theta}) \log\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} d\boldsymbol{\theta} \tag{4-60}$$

To illustrate this process, Liu & Wang (2016) first establish that

$$KL\big(q_{[\boldsymbol{T}]}||p\big) = KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) \tag{4-61}$$

which we have re-derived in *Appendix 5: Change of variables in the KLD*. Plugging in the definition of the Kullback-Leibler divergence for the RHS, we obtain:

$$KL\big(q_{[\boldsymbol{T}]}||p\big) = \int q(\boldsymbol{\theta}) \log\frac{q(\boldsymbol{\theta})}{p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta})} d\boldsymbol{\theta} \tag{4-62}$$

Now we take the derivative of both sides with respect to the transformation increment ε:

$$\nabla_\varepsilon KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) = \nabla_\varepsilon \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta})} d\boldsymbol{\theta} \tag{4-63}$$

We can split the fraction in the logarithm to obtain:

$$\nabla_\varepsilon KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) = \nabla_\varepsilon \int q(\boldsymbol{\theta}) \log q(\boldsymbol{\theta})\, d\boldsymbol{\theta} - \nabla_\varepsilon \int q(\boldsymbol{\theta}) \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{4-64}$$

Acknowledging that the first RHS term is independent of the transformation $\boldsymbol{T}$ and thus of ε, its derivative yields zero and drops out. We may further shift the derivative operator inside the integral to the logarithm:

$$\nabla_\varepsilon KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) = -\int q(\boldsymbol{\theta}) \nabla_\varepsilon \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{4-65}$$

The integral on the RHS with $q(\boldsymbol{\theta})$ corresponds to the expectation under q, yielding:

$$\nabla_\varepsilon KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) = -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\nabla_\varepsilon \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta})\right] \tag{4-66}$$

Now in order to develop this equation further, we need to establish what $\nabla_\varepsilon \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta})$ is. Towards this end, start from the change of variables formula (*Appendix 4: Change of variables*), then apply the logarithm and ∇_ε to both sides:

$$\nabla_\varepsilon \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \nabla_\varepsilon \log\left[p\left(\boldsymbol{T}(\boldsymbol{\theta})\right) \cdot \left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{T}(\boldsymbol{\theta})\right)\right|\right] \tag{4-67}$$

For the derivative of a logarithm, we have the following identity:

$$\nabla \log f(\boldsymbol{\theta}) = \frac{\nabla f(\boldsymbol{\theta})}{f(\boldsymbol{\theta})} \tag{4-68}$$

So we obtain:

$$\nabla_{\varepsilon} \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \frac{\nabla_{\varepsilon}\left[p\left(\boldsymbol{T}(\boldsymbol{\theta})\right) \cdot \left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right|\right]}{p\left(\boldsymbol{T}(\boldsymbol{\theta})\right) \cdot \left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right|} \tag{4-69}$$

Recall further the product rule of derivation:

$$\nabla(fg) = \nabla f g + f \nabla g \tag{4-70}$$

So we obtain:

$$\nabla_{\varepsilon} \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \frac{\nabla_{\varepsilon} p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)\left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right| + p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)\nabla_{\varepsilon}\left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right|}{p\left(\boldsymbol{T}(\boldsymbol{\theta})\right) \cdot \left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right|} \tag{4-71}$$

Separating RHS into two terms and cancelling redundant terms yields:

$$\nabla_{\varepsilon} \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \frac{\nabla_{\varepsilon} p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)}{p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)} + \frac{\nabla_{\varepsilon}\left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right|}{\left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right|} \tag{4-72}$$

Applying the chain rule of differentiation to the first RHS numerator, we obtain:

$$\nabla_{\varepsilon} \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \frac{\boldsymbol{\nabla}_{\boldsymbol{\theta}} p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)^{\top} \nabla_{\varepsilon}\boldsymbol{T}(\boldsymbol{\theta})}{p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)} + \frac{\nabla_{\varepsilon}\left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right|}{\left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right|} \tag{4-73}$$

Now if we use the derivative of a logarithm identity (Equation 4-68) in reverse, we obtain:

$$\nabla_{\varepsilon} \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)^{\top} \nabla_{\varepsilon}\boldsymbol{T}(\boldsymbol{\theta}) + \nabla_{\varepsilon} \log\left|\det\left(\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)\right| \tag{4-74}$$

We can use the identity $\log\det A = \operatorname{trace}\log A$ for the second RHS term:

$$\nabla_{\varepsilon} \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)^{\top} \nabla_{\varepsilon}\boldsymbol{T}(\boldsymbol{\theta}) + \nabla_{\varepsilon} \operatorname{trace}[\log \boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})] \tag{4-75}$$

Using the logarithm identity (Equation 4-68) and pulling ∇_{ε} into the trace, we obtain:

$$\nabla_{\varepsilon} \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p\left(\boldsymbol{T}(\boldsymbol{\theta})\right)^{\top} \nabla_{\varepsilon}\boldsymbol{T}(\boldsymbol{\theta}) + \operatorname{trace}\left[\nabla_{\varepsilon}\boldsymbol{\nabla}_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\left(\nabla_{\boldsymbol{\theta}}\boldsymbol{T}(\boldsymbol{\theta})\right)^{-1}\right] \tag{4-76}$$

Using the commutativity of matrices inside a trace ($\operatorname{trace}\boldsymbol{AB} = \operatorname{trace}\boldsymbol{BA}$) (Pedersen et al., 2008), we can reorder the second RHS term:

$$\nabla_\varepsilon \log p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{\theta}) = \boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p\big(\boldsymbol{T}(\boldsymbol{\theta})\big)^\top \nabla_\varepsilon \boldsymbol{T}(\boldsymbol{\theta}) + \operatorname{trace}\left[\big(\nabla_{\boldsymbol{\theta}} \boldsymbol{T}(\boldsymbol{\theta})\big)^{-1} \nabla_\varepsilon \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{T}(\boldsymbol{\theta})\right] \quad 4\text{-}77$$

Now we can plug this into Equation 4-66:

$$\begin{aligned} &\nabla_\varepsilon KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) \\ &\qquad = -\mathbb{E}_{\boldsymbol{x}\sim q}\left[\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p\big(\boldsymbol{T}(\boldsymbol{\theta})\big)^\top \nabla_\varepsilon \boldsymbol{T}(\boldsymbol{\theta})\right. \\ &\qquad \left. + \operatorname{trace}\left[\big(\nabla_{\boldsymbol{\theta}} \boldsymbol{T}(\boldsymbol{\theta})\big)^{-1} \nabla_\varepsilon \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{T}(\boldsymbol{\theta})\right]\right] \end{aligned} \quad 4\text{-}78$$

Note that when $\boldsymbol{T}(\boldsymbol{\theta}) = \boldsymbol{\theta} + \varepsilon \boldsymbol{\phi}(\boldsymbol{\theta})$ we have the following identities in the limit of $\varepsilon = 0$:

$$\boldsymbol{T}(\boldsymbol{\theta})|_{\varepsilon=0} = \boldsymbol{\theta} \quad 4\text{-}79$$

$$\nabla_\varepsilon \boldsymbol{T}(\boldsymbol{\theta})|_{\varepsilon=0} = \boldsymbol{\phi}(\boldsymbol{\theta}) \quad 4\text{-}80$$

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{T}(\boldsymbol{\theta})|_{\varepsilon=0} = \boldsymbol{I} \quad 4\text{-}81$$

$$\nabla_\varepsilon \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{T}(\boldsymbol{\theta})|_{\varepsilon=0} = \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{\phi}(\boldsymbol{\theta}) \quad 4\text{-}82$$

where $\boldsymbol{I}$ is the identity matrix, whose inverse is also the identity matrix. Plugging these identities into Equation 4-78 and omitting the inverse identity matrix yields:

$$\nabla_\varepsilon KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) = -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta})^\top \boldsymbol{\phi}(\boldsymbol{\theta}) + \operatorname{trace} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{\phi}(\boldsymbol{\theta})\right]|_{\varepsilon=0} \quad 4\text{-}83$$

Using the trace identity (Equation 4-44) and pulling the trace operator outside, we obtain:

$$\nabla_\varepsilon KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) = -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\operatorname{trace}\left[\boldsymbol{\nabla}_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta})^\top \boldsymbol{\phi}(\boldsymbol{\theta}) + \boldsymbol{\nabla}_{\boldsymbol{\theta}} \boldsymbol{\phi}(\boldsymbol{\theta})\right]\right]|_{\varepsilon=0} \quad 4\text{-}84$$

Using Equation 4-11 and multiplying both sides by -1, we can simplify this to:

$$\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\operatorname{trace}\left[A_p \boldsymbol{\phi}(\boldsymbol{\theta})\right]\right]|_{\varepsilon=0} = -\nabla_\varepsilon KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) \quad 4\text{-}85$$

This result establishes the connection between Stein discrepancy and the Kullback-Leibler divergence. As such, the vector field $\boldsymbol{\phi}^*(\boldsymbol{\theta})$ which maximizes the KSD corresponds to the steepest descent direction of the Kullback-Leibler divergence.

4.13 Appendix 4: Change of variables

Assume a transformation $\boldsymbol{T}: \mathbb{R}^d \to \mathbb{R}^d$ and $\boldsymbol{x} \in \mathbb{R}^d$. Denote by $\boldsymbol{T}(\boldsymbol{x})$ the values of $\boldsymbol{x}$ after the transformation, by q the distribution after the transformation, and by $q_{[\boldsymbol{T}^{-1}]}$ the distribution before the transformation. Throughout the transformation probability mass is conserved, so we have:

$$\int q_{[\boldsymbol{T}^{-1}]}(\boldsymbol{x})d\boldsymbol{x} = \int q\big(\boldsymbol{T}(\boldsymbol{x})\big)d\boldsymbol{T}(\boldsymbol{x}) \tag{4-86}$$

We want to express the right-hand side (RHS) in terms of $\boldsymbol{x}$, so we expand it by adding $d\boldsymbol{x}/d\boldsymbol{x}$:

$$\int q_{[\boldsymbol{T}^{-1}]}(\boldsymbol{x})d\boldsymbol{x} = \int q\big(\boldsymbol{T}(\boldsymbol{x})\big)\frac{d\boldsymbol{T}(\boldsymbol{x})}{d\boldsymbol{x}}d\boldsymbol{x} \tag{4-87}$$

This suggests that the arguments of the two integrals are equal:

$$q_{[\boldsymbol{T}^{-1}]}(\boldsymbol{x}) = q\big(\boldsymbol{T}(\boldsymbol{x})\big)\frac{d\boldsymbol{T}(\boldsymbol{x})}{d\boldsymbol{x}} \tag{4-88}$$

which is equivalent to the desired result:

$$q_{[\boldsymbol{T}^{-1}]}(\boldsymbol{x}) = q\big(\boldsymbol{T}(\boldsymbol{x})\big) \cdot \left|\det\big(\nabla_{\boldsymbol{x}}\boldsymbol{T}(\boldsymbol{x})\big)\right| \tag{4-89}$$

or inversely:

$$q\big(\boldsymbol{T}(\boldsymbol{x})\big) = q_{[\boldsymbol{T}^{-1}]}(\boldsymbol{x})\frac{d\boldsymbol{x}}{d\boldsymbol{T}(\boldsymbol{x})} = q_{[\boldsymbol{T}^{-1}]}(\boldsymbol{x})\left|\det\left(\nabla_{\boldsymbol{x}}\boldsymbol{T}^{-\mathbf{1}}(\boldsymbol{x})\right)\right| \tag{4-90}$$

where $\boldsymbol{T}^{-\mathbf{1}}$ is the inverse of $\boldsymbol{T}$.

4.14 Appendix 5: Change of variables in the KLD

Assume a transformation $\boldsymbol{T}: \mathbb{R}^d \to \mathbb{R}^d$ and $\boldsymbol{x} \in \mathbb{R}^d$. Denote by $\boldsymbol{T}(\boldsymbol{x})$ the values of $\boldsymbol{x}$ after the transformation, by $q_{[\boldsymbol{T}]}$ the distribution after, and by q the distribution before the transformation. $p\big(\boldsymbol{T}(\boldsymbol{x})\big)$ is the probability density of p at transformed position $\boldsymbol{T}(\boldsymbol{x})$. With this, let us define the Kullback-Leibler divergence between distributions $q_{[\boldsymbol{T}]}$and p:

$$KL\big(q_{[\boldsymbol{T}]}||p\big) = \int q_{[\boldsymbol{T}]}\big(\boldsymbol{T}(\boldsymbol{x})\big) \log \frac{q_{[\boldsymbol{T}]}\big(\boldsymbol{T}(\boldsymbol{x})\big)}{p\big(\boldsymbol{T}(\boldsymbol{x})\big)} d\boldsymbol{T}(\boldsymbol{x}) \qquad 4\text{-}91$$

Using the results of *Appendix 4: Change of variables*, we obtain:

$$KL\big(q_{[\boldsymbol{T}]}||p\big) = \int q(\boldsymbol{x}) \frac{d\boldsymbol{x}}{d\boldsymbol{T}(\boldsymbol{x})} \log \frac{q(\boldsymbol{x}) \frac{d\boldsymbol{x}}{d\boldsymbol{T}(\boldsymbol{x})}}{p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{x}) \frac{d\boldsymbol{x}}{d\boldsymbol{T}(\boldsymbol{x})}} d\boldsymbol{T}(\boldsymbol{x}) \qquad 4\text{-}92$$

noting that we use a different subscript notation than in *Appendix 4: Change of variables* (from the perspective of the original distribution rather than the transformed one, but following the convention that $\boldsymbol{T}: p_{[\boldsymbol{T}^{-1}]} \to p \; or \; p \to p_{[\boldsymbol{T}]}$). Noting that the Jacobian determinants in the logarithm's argument cancel out, and shifting the Jacobian determinant to the end and cancelling out $d\boldsymbol{T}(\boldsymbol{x})$, we obtain the desired identity:

$$KL\big(q_{[\boldsymbol{T}]}||p\big) = \int q(\boldsymbol{x}) \log \frac{q(\boldsymbol{x})}{p_{[\boldsymbol{T}^{-1}]}(\boldsymbol{x})} d\boldsymbol{x} = KL\left(q||p_{[\boldsymbol{T}^{-1}]}\right) \qquad 4\text{-}93$$

5 Concluding Discussion

The efficient implementation of non-linear, non-Gaussian Bayesian parameter inference remains one of the largest scientific challenges in hydrogeology and beyond. In this dissertation, I investigated and adapted a number of techniques which hold promise in this endeavour.

5.1 Challenges and solutions

This dissertation presented two studies (Chapter 2, Chapter 3) exploring variations of the particle filter framework. Filter methods harbour great potential for sequential, online parameter inference. Their generality renders them theoretically capable of approximating arbitrary distributions, and their algorithmic structure lends itself naturally to sequential parameter inference, which is of great interest for self-improving groundwater models. Their limitations, unfortunately, lie in particle degeneracy and eventual ensemble collapse. To address these limitations, a number of promising directions could be explored:

1. **Alternative likelihood formulations**, particularly those with heavier tails (*higher probability of outliers*) or auto-correlation[18] (*higher probability of recurring deviations*) can reduce the divergence of the likelihood and thus slow the tapering of the posterior.
2. **Information exchange across the ensemble** could be used to direct the rejuvenation mechanisms and make its exploration of

[18] Positively spatially and/or temporally correlated errors assume that if one prediction deviates in a certain way from the observations – for example by under-prediction – other nearby predictions (in space or time) will display similar deviations. As a practical example: if a groundwater model over-predicts the water table in a well at a certain time by 2 m, chances are high that the next prediction 15 minutes later will display a similar error. A temporally correlated error would reflect this.

parameter space more efficient. Proposals based on mutation operations of genetic algorithms, for example, mimic the optimization benefits of sexual reproduction in biological evolution[19]. Estimating approximate gradients of the posterior distribution from other ensemble members can further inform promising mutation directions.

3. **Pragmatic workarounds** based on variance inflation techniques are an efficient yet dangerous option. The addition of entropy through artificial error components (e.g., Chapter 2) can slow and even reverse particle degeneracy, but fades the ensemble's statistical memory and corrupts the inference problem if used without theoretical justification[20]. If such methods are used, it is important to find ways to limit the error growth.

Particle filters seem particularly well-suited for *state inference*, the uncertainty estimation for transient variables such as hydraulic heads, contaminant concentrations, or temperatures. This can be useful in certain scenarios – for example the real-time control of pumping wells (Bauser et al., 2010) – but in general the interest lies in the inference of subsurface properties. Extending the inference to time-invariable parameters is not always straightforward.

[19] As opposed to classic particle filters, or equivalently asexual reproduction, beneficial mutations no longer have to occur within a single lineage. Beneficial mutations of multiple individuals can be explored in parallel, and combined into a new individual, shortening the evolution process. Without information exchange between the particles, we are restricted to single lineages.

[20] An example of a justified error: explicit forecast noise corrupts knowledge of transient variables during forecast and constitute a form of variance inflation. In this case, this is desired, as it reflects the imperfection of the numerical model as a vehicle for the transformation of information: in time, the noise may override the model predictions, blurring the forecast into an ambiguous fog. A time-invariable piece of knowledge, however, such as the identity of the culprit in the murder case, should neither change nor corrupt with time.

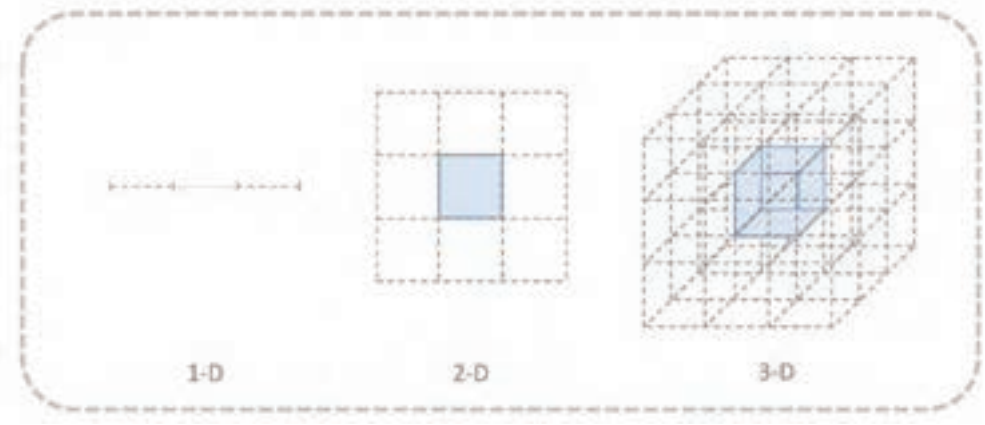


Figure 5-1. Illustration of the curse of dimensionality, adapted from Betancourt (2018). In a one-dimensional space, the centre stretch (blue) occupies $1/3$ of its neighbourhood. In two dimensions, this reduces to $1/9$, and in three dimensions to $1/27$. For higher-dimensional spaces, there are generally $d^3 - 1$ neighbouring partitions in a d-dimensional space. This fraction quickly becomes negligibly small with higher dimensions.

The primary issue of particle filters lies in the curse of dimensionality, a geometric property of high-dimensional spaces (Figure 5-1). This issue is shared with other methods which rely on random mutations for the exploration of parameter space (such as MCMC): the larger the dimensionality of the space, the lower the probability to jump in a direction which constitutes an improvement, or to land in a specific region of parameter space, such as the optimum. In the extremely high-dimensional spaces prevailing in hydrogeology, this chance is almost negligibly small.

It is an interesting property of the Ensemble Kalman Filter (Evensen, 1994, 2003) that it does not appear to share this weakness in practical applications, despite seemingly operating in the same space with the same properties. An explanation may be found in its theoretical foundation, which approximates its Gaussian's covariance matrix with Monte Carlo estimates taken from the particle ensemble. If the number of particles N minus one is smaller than the dimensionality of parameter space D ($N - 1 < D$), this covariance matrix must be rank-deficient. In other words: the approximated covariance matrix lives only on an at-

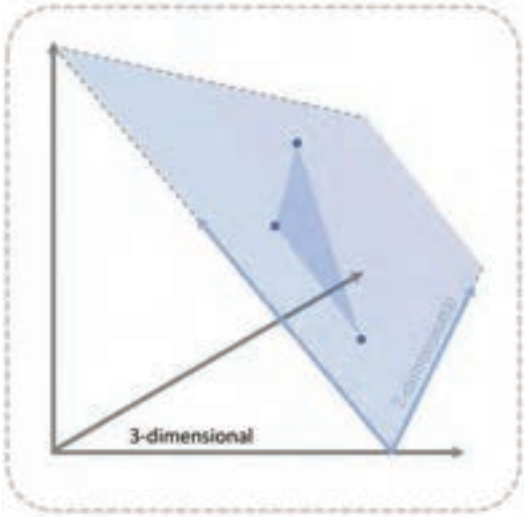


Figure 5-2. Illustration of a two-dimensional ensemble-based subspace for $N = 3$ and $D = 3$. All points accessible through linear combinations of the three particles lie in a two-dimensional plane (shaded blue).

most $N-1$ -dimensional subspace of the embedding D-dimensional parameter space (Figure 5-2), which effectively hides the true dimensionality of parameter space from the algorithm.

This is computationally convenient, because it implicitly scales the complexity of the inference problem with the computational resources available (in our case: the ensemble size). Unfortunately, this is a devil's bargain: the price for this efficiency is that the algorithm can only explore the subspace, whose dimensions are comprised of specific parameter combinations and defined by where the prior particles are located. If the true optima are located outside this subspace, the inference algorithm will not be able to find them. Nonetheless – following the virtue of pragmatism – one might argue that an approximate, potentially flawed but obtainable uncertainty estimate is to be preferred over a mathematically perfect unobtainable one.

A further argument can be made that the high dimensionality encountered in many hydrogeological models is mostly 'phantom dimensionality'. The parameter count, which can sometimes be in the millions, is primarily owed to the fine grid discretization required to represent non-trivial, spatial heterogeneities [21]. In practice, the

[21] A parallel could be drawn to computer screens: To visualize variations of an image of – say – an apple, only a single parameter (say, its location on the screen) might suffice. If we want to represent arbitrary images, however, we require additional parameters, and eventually it makes

parameters of nearby cells are likely to be heavily correlated[22]. As a consequence, a million-dimensional parameter space may have an effective dimensionality which is only in the tens or hundreds. This effective dimensionality can be quantified by filling a Jacobian matrix[23], taking its singular value decomposition, and seeing how many of the parameter space's singular vectors have significantly nonzero singular values.

This leaves the question of how we could mimic this subspace-scaling property for non-Gaussian parameter inference. Chapter 4 explored the use of Stein Variational Gradient Descent (Liu & Wang, 2016), which lends itself naturally to multi-modal distributions as long as they are sufficiently smooth. Where cheap evaluations of the model's Jacobian matrix or the log-posterior gradient are available, SVGD need not even confine itself to a subspace. In practice, however, neither the log-posterior gradient nor the Jacobian will generally be available in closed form, and estimating it with finite differences in high-dimensional spaces is an exercise which quickly becomes computationally unfeasible.

In such scenarios, it may be necessary to employ ensemble-based approximations of the Jacobian matrix (see Chapter 4.4.2). Since these ensemble-based approximations only exploit gradients between the available particles, the resulting Jacobian will also generally be rank-deficient and confined to an at most $N - 1$-dimensional subspace. This

more sense to represent the image as pixels. While each single image we visualize will never require the full pixel resolution, the full dimensionality is nonetheless required for the versatility.

[22] A-priori simplification may not always be possible, since the nature of this correlation structure is often non-trivial, particularly in pursuit of geological realism (see Chapter 3).

[23] Jacobians are essentially model diagnostic matrices: they contain detailed predictive information about how the model's output will respond to changes to its parameters. Needless to say, this information is extremely useful if you want to optimize a model's parameters to achieve a desired output.

property mimics the subspace scaling of the EnKF, with all its advantages and drawbacks, and could render SVGD an interesting direction for further research. Promising directions to improve its use might be:

1. **Alternative Jacobian or log-posterior gradient approximations**, as either constitutes a key ingredient to the algorithm, and neither is usually available in closed form. As we explored in Chapter 4, a number of different approaches can be used, and a more comprehensive comparison of their strengths and weaknesses might be a valuable addition. While ensemble-based approximations are a very interesting non-intrusive option, Jacobians obtained from Automatic Differentiation seem an even better option, since their rank is independent of the ensemble size.
2. **Alternative gradient descent algorithms** could significantly improve the computational efficiency of the algorithm. SVGD only provides the descent direction and transformation function, and can thus interface with different descent algorithms which adjust the step size, such as ADAM (Kingma & Ba, 2015). As in other gradient descent applications, the optimal choice of algorithm depends on the model.

Already now, SVGD seems a highly promising avenue for practical application in hydrogeological modelling. I remain cautiously optimistic that it could one day rival or replace the ubiquity of the EnKF, although a rigorous comparison of the two methods, and a better understanding of the optimization performance of SVGD still remains an open task.

Beyond the methods explored in this dissertation lie further approaches which could hold potential for hydrogeological application. In this

dissertation, I primarily explored sequential or iterative inference methods, and only grazed the topic of direct samplers in Chapter 3. These two fields approach the subject of inference in slightly different ways:

- **Direct samplers** like MCMC sample the posterior distribution – which need not be available in analytic form – directly. In this case, the Bayesian inference part (use of Bayes' theorem) is evaluated individually, analytically, and pointwise. We then construct the posterior ensemble piece by piece.
- **Sequential or iterative inference methods**, such as the ones explored in this dissertation, start off from an initial ensemble and then gradually transform it into a posterior ensemble. This is done either through the gradual addition of information (filter techniques), or through iterative transformation (SVGD). In both cases, the posterior ensemble is obtained through individual particle transformation.

Apart from these approaches, however, lies a third path. This path is related to the method explored in Chapter 4:

- **Transportation methods** attempt to explicitly reconstruct the transformation function which converts the prior into the posterior (e.g., Marzouk et al., 2017; Spantini et al., 2018, 2019).

Instead of considering the transformation as a means to an end, transportation method focus on identifying the (ideally global) nature of the change itself. The advantages are evident: If we had a sufficiently faithful representation of the transformation function, we could draw as many posterior samples as desired with only negligible computational effort. This could be achieved by first sampling the prior (or a reference

distribution), then transforming the sample into a posterior sample. This contrasts direct sampling (for which each new sample comes with significant computational cost) or sequential inference (which starts off with a certain ensemble size which is not trivial to adjust later). At the moment, these methods seem to have found only limited application outside of theoretical scenarios, so an exploration of their strengths and limitations in practical contexts would be one of many highly interesting directions for future research.

5.2 Frontiers old and new

We live in a fascinating time for science. With the growth of computational power and predictive fidelity, numerical models find their way into ever greater aspects of modern life. As our reliance on such systematic representations of our understanding of the world increases, so does our need to identify their limitations and ambiguities, to create new foundations on which the next generations of models will be built.

Algorithmic uncertainty estimation will play an important part in this. Rigorous Bayesian inference may partially lift the task of judging the plausibility of different hypotheses off our shoulders, to be shared with algorithms less burdened by intuitive preconceptions.

Much time has passed since Nicolaus Copernicus has recorded his model of the universe. More time still since the ancient Babylonians first carved their model of the universe into tablets of clay. The iterations of our understanding of the world continue, but our tools have evolved. Where ancient lawgivers like Hammurabi had to chisel their laws into stone, trading whim for reliability and personal judgement for justice, modern algorithms may not only represent the laws through which we judge

plausibility, but also apply them – and even alert us to possibilities we might not have considered. And maybe one day, the next great scientific breakthrough will not be discovered by a human, but an algorithm.

6 References

Aanonsen, S. I., Nævdal, G., Oliver, D. S., Reynolds, A. C., & Vallès, B. (2009). The Ensemble Kalman Filter in Reservoir Engineering--a Review. *SPE Journal*, *14*(03), 393–412. https://doi.org/10.2118/117274-PA

Abbaszadeh, P., Moradkhani, H., & Yan, H. (2018). Enhancing hydrologic data assimilation by evolutionary Particle Filter and Markov Chain Monte Carlo. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2017.11.011

Abdoun, O., Abouchabaka, J., & Tajani, C. (2012). Analyzing the performance of mutation operators to solve the travelling salesman problem. *ArXiv Preprint ArXiv:1203.309*.

Alcolea, A., & Renard, P. (2010). Blocking Moving Window algorithm: Conditioning multiple-point simulations to hydrogeological data. *Water Resources Research*. https://doi.org/10.1029/2009WR007943

Amezcua, J., & Van Leeuwen, P. J. (2014). Gaussian anamorphosis in the analysis step of the EnKF: a joint state-variable/observation approach. *Tellus A: Dynamic Meteorology and Oceanography*. https://doi.org/10.3402/tellusa.v66.23493

Anderson, J. L. (2007). An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus, Series A: Dynamic Meteorology and Oceanography*. https://doi.org/10.1111/j.1600-0870.2006.00216.x

Anderson, J. L., & Anderson, S. L. (1999). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*. https://doi.org/10.1175/1520-0493(1999)127<2741:AMCIOT>2.0.CO;2

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian\nBayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188. https://doi.org/10.1109/78.978374

Aurenhammer, F., Klein, R., & Lee, D. T. (2013). *Voronoi diagrams and delaunay triangulations*. *Voronoi Diagrams and Delaunay Triangulations*. https://doi.org/10.1142/8685

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. *Second International Conference on Genetic Algorithms and Their Application*, *206*, 260.

https://doi.org/10.1136/bmj.39184.617049.80

Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J., & Fienen, M. N. (2016). Scripting MODFLOW Model Development Using Python and FloPy. *Groundwater*, *54*(5), 733–739. https://doi.org/10.1111/gwat.12413

Barrash, W., Clemo, T., Fox, J. J., & Johnson, T. C. (2006). Field, laboratory, and modeling investigation of the skin effect at wells with slotted casing, Boise Hydrogeophysical Research Site. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2005.10.029

Bauser, G., Franssen, H. J. H., Kaiser, H. P., Kuhlmann, U., Stauffer, F., & Kinzelbach, W. (2010). Real-time management of an Urban groundwater well field threatened by pollution. *Environmental Science and Technology*. https://doi.org/10.1021/es100648j

Bengtsson, T., Bickel, P., & Li, B. (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and Statistics: Essays in Honor of David A. Freedman*. https://doi.org/10.1214/193940307000000518

Bennett, J. P., Haslauer, C. P., Ross, M., & Cirpka, O. A. (2019). An Open, Object-Based Framework for Generating Anisotropy in Sedimentary Subsurface Models. *Groundwater*. https://doi.org/10.1111/gwat.12803

Betancourt, M. (2018). A Conceptual Introduction to Hamiltonian Monte Carlo. arXiv. Retrieved from https://arxiv.org/abs/1701.02434

Burri, N. M., Weatherl, R., Moeck, C., & Schirmer, M. (2019). A review of threats to groundwater quality in the anthropocene. *Science of the Total Environment*. https://doi.org/10.1016/j.scitotenv.2019.05.236

Caers, J., & Zhang, T. (2005). Multiple-point geostatistics: A quantitative vehicle for integrating geologic analogs into multiple reservoir models. *AAPG Memoir*.

Caers, J., Strebelle, S., & Payrazyan, K. (2003). Stochastic integration of seismic data and geologic scenarios: A West Africa submarine channel saga. *The Leading Edge*. https://doi.org/10.1190/1.1564521

Cardell-Oliver, R., Kranz, M., Smettem, K., & Mayer, K. (2005). A Reactive Soil Moisture Sensor Network: Design and Field Evaluation. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1080/15501320590966422

Carrera, Jesus, & Neuman, S. P. (1986). Estimation of Aquifer Parameters Under Transient and Steady State Conditions: 1. Maximum Likelihood Method Incorporating Prior Information. *Water*

Resources Research. https://doi.org/10.1029/WR022i002p00199

Carrera, Jesús, Alcolea, A., Medina, A., Hidalgo, J., & Slooten, L. J. (2005). Inverse problem in hydrogeology. *Hydrogeology Journal*. https://doi.org/10.1007/s10040-004-0404-7

Chen, Y., & Oliver, D. S. (2013). Levenberg-Marquardt forms of the iterative ensemble smoother for efficient history matching and uncertainty quantification. *Computational Geosciences*. https://doi.org/10.1007/s10596-013-9351-5

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*. https://doi.org/10.1093/biomet/89.3.539

Chopin, N., Jacob, P. E., & Papaspiliopoulos, O. (2013). SMC2: An efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*. https://doi.org/10.1111/j.1467-9868.2012.01046.x

Chwialkowski, K., Strathmann, H., & Gretton, A. (2016). A kernel test of goodness of fit. In *33rd International Conference on Machine Learning, ICML 2016*.

Cirpka, O. A., & Valocchi, A. J. (2016). Debates—Stochastic subsurface hydrology from theory to practice: Does stochastic subsurface hydrology help solving practical problems of contaminant hydrogeology? *Water Resources Research*. https://doi.org/10.1002/2016WR019087

Crestani, E., Camporese, M., Baú, D., & Salandin, P. (2013). Ensemble Kalman filter versus ensemble smoother for assessing hydraulic conductivity via tracer test data assimilation. *Hydrology and Earth System Sciences*. https://doi.org/10.5194/hess-17-1517-2013

Crisan, D., & Miguez, J. (2018). Nested particle filters for online parameter estimation in discrete-time state-space Markov models. *Bernoulli*, *24*(4A), 3039--3086.

Crisan, Dan, & Miguez, J. (2013). Nested particle filters for online parameter estimation in discrete-time state-space Markov models. Retrieved from http://arxiv.org/abs/1308.1883

D'Agata, I. (2019). *Italy's Native Wine Grape Terroirs*. Univ of California Press.

Deutsch, C. V., & Tran, T. T. (2002). FLUVSIM: A program for object-based stochastic modeling of fluvial depositional systems. *Computers and Geosciences*. https://doi.org/10.1016/S0098-3004(01)00075-9

Doherty, J, & Hunt, R. (2010). Approaches to highly parameterized

inversion: a guide to using PEST for groundwater-model calibration. *U. S. Geological Survey Scientific Investigations Report 2010-5169*. https://doi.org/2010-5211

Doherty, John. (2015). *Calibration and Uncertainty Analysis for Complex Environmental Models*. Brisbane, Australia: Watermark Numerical Computing.

Doherty, John, & Christensen, S. (2011). Use of paired simple and complex models to reduce predictive bias and quantify uncertainty. *Water Resources Research*. https://doi.org/10.1029/2011WR010763

Doherty, John, Fienen, M. N., & Hunt, R. J. (2010). *Approaches to Highly Parameterized Inversion : Pilot-Point Theory , Guidelines , and Research Directions*. *Consume hasta Morir*. US Geological Survey. https://doi.org/2010-5168

Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*.

Doucet, A., & Tadić, V. B. (2003). Parameter estimation in general state-space models using particle methods. In *Annals of the Institute of Statistical Mathematics*. https://doi.org/10.1023/A:1026390323638

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*.

Evans, J., & Jones, A. (2000). The History and Practice of Ancient Astronomy. *American Journal of Physics*. https://doi.org/10.1119/1.19412

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, *99*(C5), 10,110-143,162. https://doi.org/10.1029/94JC00572

Evensen, G. (2003). The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, *53*(4), 343–367. https://doi.org/10.1007/s10236-003-0036-9

Farchi, A., & Bocquet, M. (2018). Review article: Comparison of local particle filters and new implementations. *Nonlinear Processes in Geophysics*. https://doi.org/10.5194/npg-25-765-2018

Fogg, G. E., & Zhang, Y. (2016). Debates—Stochastic subsurface hydrology from theory to practice: A geologic perspective. *Water Resources Research*. https://doi.org/10.1002/2016WR019699

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013). emcee : The MCMC Hammer . *Publications of the Astronomical Society*

of the Pacific. https://doi.org/10.1086/670067

Fraser, C. G. (2006). *The cosmos: a historical perspective* (1.). Westport, Connecticut: Greenwood Publishing Group, Inc.

Geppert, G. (2015). *Analysis and application of the ensemble Kalman filter for the estimation of bounded quantities*. Hamburg. Retrieved from https://www.mpimet.mpg.de/fileadmin/publikationen/Reports/WEB_BzE_168.pdf

Glover, W., & Lygeros, J. (2004). A stochastic hybrid model for air traffic control simulation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian Bayesian state estimation. *IEE Proceedings, Part F: Radar and Signal Processing*. https://doi.org/10.1049/ip-f-2.1993.0015

Gorelick, S. M., & Zheng, C. (2015). Global change and the groundwater management challenge. *Water Resources Research*. https://doi.org/10.1002/2014WR016825

Gu, Y., & Oliver, D. S. (2007). An Iterative Ensemble Kalman Filter for Multiphase Fluid Flow Data Assimilation. *SPE Journal*, *12*(4), 438–446. https://doi.org/10.2118/108438-pa

Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2009.08.003

Hamill, T. M., Whitaker, J. S., & Snyder, C. (2001). Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*. https://doi.org/10.1175/1520-0493(2001)129<2776:DDFOBE>2.0.CO;2

Hendricks Franssen, H. J., & Kinzelbach, W. (2008). Real-time groundwater flow modeling with the Ensemble Kalman Filter: Joint estimation of states and parameters and the filter inbreeding problem. *Water Resources Research*, *44*(9). https://doi.org/10.1029/2007WR006505

Hendricks Franssen, H. J., Alcolea, A., Riva, M., Bakr, M., van der Wiel, N., Stauffer, F., & Guadagnini, A. (2009). A comparison of seven methods for the inverse modelling of groundwater flow. Application to the characterisation of well catchments. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2009.02.011

Hendricks Franssen, H. J., Kaiser, H. P., Kuhlmann, U., Bauser, G., Stauffer, F., Mller, R., & Kinzelbach, W. (2011). Operational real-time modeling with ensemble Kalman filter of variably saturated subsurface flow including stream-aquifer interaction and parameter updating. *Water Resources Research*. https://doi.org/10.1029/2010WR009480

Hill, M. C., Banta, E. R., Harbaugh, A. W., & Alderman, E. R. (2000). MODFLOW-2000, the U.S. Geological Survey Modular Ground-Water Model—User Guide To the Observation, Sensitivity, and Parameter-Estimation Processes and Three Post-Processing Programs. *U.S. Geological Survey Open-File Report 00-184*.

Holmes, C., Krzystof, L., & Pompe, E. (2017). *Adaptive MCMC for multimodal distributions*. Retrieved from https://pdfs.semanticscholar.org/c75d/f035c23e3c0425409e70d457cd43b174076f.pdf

Houser, P. R., Shuttleworth, W. J., Famiglietti, J. S., Gupta, H. V., Syed, K. H., & Goodrich, D. C. (1998). Integration of soil moisture remote sensing and hydrologic modeling using data assimilation. *Water Resources Research*. https://doi.org/10.1029/1998WR900001

Hu, L. Y., Zhao, Y., Liu, Y., Scheepens, C., & Bouchard, A. (2013). Updating multipoint simulations using the ensemble Kalman filter. *Computers & Geosciences*, *51*, 7–15. https://doi.org/10.1016/j.cageo.2012.08.020

Iglesias, M. A., Law, K. J. H., & Stuart, A. M. (2013). Ensemble kalman methods for inverse problems. *Inverse Problems*, *29*(4). https://doi.org/10.1088/0266-5611/29/4/045001

Jafarpour, B., & McLaughlin, D. B. (2009). *Estimating Channelized-Reservoir Permeabilities With the Ensemble Kalman Filter: The Importance of Ensemble Design*. *SPE Journal* (Vol. 14). https://doi.org/10.2118/108941-pa

Jafarpour, B., & Tarrahi, M. (2011). Assessing the performance of the ensemble Kalman filter for subsurface flow data integration under variogram uncertainty. *Water Resources Research*, *47*(5), 1–16. https://doi.org/10.1029/2010WR009090

Journel, A., & Zhang, T. (2006). The necessity of a multiple-point prior model. *Mathematical Geology*. https://doi.org/10.1007/s11004-006-9031-2

Katzfuss, M., Stroud, J. R., & Wikle, C. K. (2016). Understanding the Ensemble Kalman Filter. *The American Statistician*, *70*(4), 350–357.

https://doi.org/10.1080/00031305.2016.1141709

Keller, J., Hendricks Franssen, H. J., & Marquart, G. (2018). Comparing Seven Variants of the Ensemble Kalman Filter: How Many Synthetic Experiments Are Needed? *Water Resources Research*. https://doi.org/10.1029/2018WR023374

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Kitagawa, G. (1998). A Self-Organizing State-Space Model. *Journal of the American Statistical Association*. https://doi.org/10.2307/2669862

Kresic, N., & Stevanovic, Z. (2010). *Groundwater Hydrology of Springs*. *Groundwater Hydrology of Springs*. https://doi.org/10.1016/C2009-0-19145-6

Kruschke, J. (2015). *Doing Bayesian data analysis: A tutorial introduction with R JAGS, and Stan*. *Igarss 2014*. https://doi.org/10.1002/9781444395105.fmatter

Kumar, N., Hendriks, B. S., Janes, K. A., de Graaf, D., & Lauffenburger, D. A. (2006). Applying computational modeling to drug discovery and development. *Drug Discovery Today*. https://doi.org/10.1016/j.drudis.2006.07.010

Langevin, C. D., Hughes, J. D., Banta, E. R., Niswonger, R. G., Panday, S., & Provost, A. M. (2017). Documentation for the MODFLOW 6 Groundwater Flow Model. *U.S. Geological Survey*. https://doi.org/10.3133/tm6A55

Lee, C.-Y., & Antonsson, E. K. (2000). Variable Length Genomes for Evolutionary Algorithms. In *Genetic and Evolutionary Computation Conference (GECCO-2000)*.

van Leeuwen, P. J. (2003). A variance-minimizing filter for large-scale applications. *Monthly Weather Review*. https://doi.org/10.1175/1520-0493(2003)131<2071:AVFFLA>2.0.CO;2

van Leeuwen, P. J. (2009). Particle Filtering in Geophysical Systems. *Monthly Weather Review*. https://doi.org/10.1175/2009MWR2835.1

van Leeuwen, P. J. (2010). Nonlinear data assimilation in geosciences: An extremely efficient particle filter. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.699

van Leeuwen, P. J. (2015). Representation errors and retrievals in linear and nonlinear data assimilation. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.2464

van Leeuwen, P. J., Künsch, H. R., Nerger, L., Potthast, R., & Reich, S. (2019). Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.3551

Li, T., Sun, S., Sattar, T. P., & Corchado, J. M. (2014). Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2013.12.031

Li, T., Bolić, M., & Djurić, P. M. (2015). Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *IEEE Signal Processing Magazine*. https://doi.org/10.1109/MSP.2014.2330626

Linde, N., Renard, P., Mukerji, T., & Caers, J. (2015). Geological realism in hydrogeological and geophysical inverse modeling: A review. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2015.09.019

Linde, N., Ginsbourger, D., Irving, J., Nobile, F., & Doucet, A. (2017). On uncertainty quantification in hydrogeology and hydrogeophysics. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2017.10.014

Liu, Q., & Wang, D. (2016). Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*.

Liu, Q., Lee, J. D., & Jordan, M. (2016). A kernelized Stein discrepancy for goodness-of-fit tests. In *33rd International Conference on Machine Learning, ICML 2016*.

Margossian, C. C. (2019). A review of automatic differentiation and its efficient implementation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. https://doi.org/10.1002/widm.1305

Mariethoz, G., Renard, P., & Straubhaar, J. (2010). The direct sampling method to perform multiple-point geostatistical simulations. *Water Resources Research*. https://doi.org/10.1029/2008WR007621

Mariethoz, G., Straubhaar, J., Renard, P., Chugunova, T., & Biver, P. (2015). Constraining distance-based multipoint simulations to proportions and trends. *Environmental Modelling and Software*. https://doi.org/10.1016/j.envsoft.2015.07.007

de Marsily, G., Delay, F., Gonçalvès, J., Renard, P., Teles, V., & Violette, S. (2005). Dealing with spatial heterogeneity. *Hydrogeology Journal*. https://doi.org/10.1007/s10040-004-0432-3

Marzouk, Y., Moselhy, T., Parno, M., & Spantini, A. (2017). Sampling via

measure transport: An introduction. In *Handbook of Uncertainty Quantification*. https://doi.org/10.1007/978-3-319-12385-1_23

Maul, S. M. (2015). *Das Gilgamesch-Epos: Neu übersetzt und kommentiert von Stefan M. Maul* (6th ed.). Munich: C.H. Beck oHG.

McLaughlin, D., & Townley, L. R. (1996). A reassessment of the groundwater inverse problem. *Water Resources Research*. https://doi.org/10.1029/96WR00160

Meteoswiss. (2020). Meteoswiss. Retrieved from www.meteoswiss.admin.ch

Moeck, C., Molson, J., & Schirmer, M. (2020). Pathline Density Distributions in a Null-Space Monte Carlo Approach to Assess Groundwater Pathways. *Groundwater*. https://doi.org/10.1111/gwat.12900

Montzka, C., Moradkhani, H., Weihermüller, L., Franssen, H. J. H., Canty, M., & Vereecken, H. (2011). Hydraulic parameter estimation by remotely-sensed top soil moisture observations with the particle filter. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2011.01.020

Moradkhani, H., Hsu, K.-L., Gupta, H., & Sorooshian, S. (2005). Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resources Research*, *41*(5). https://doi.org/10.1029/2004WR003604

Moradkhani, H., Dechant, C. M., & Sorooshian, S. (2012). Evolution of ensemble data assimilation for uncertainty quantification using the particle filter-Markov chain Monte Carlo method. *Water Resources Research*. https://doi.org/10.1029/2012WR012144

Morzfeld, M., Hodyss, D., & Snyder, C. (2017). What the collapse of the ensemble Kalman filter tells us about particle filters. *Tellus, Series A: Dynamic Meteorology and Oceanography*. https://doi.org/10.1080/16000870.2017.1283809

El Moselhy, T. A., & Marzouk, Y. M. (2012). Bayesian inference with optimal maps. *Journal of Computational Physics*. https://doi.org/10.1016/j.jcp.2012.07.022

Nemet-Nejat, K. R. (1998). *Daily life in ancient Mesopotamia* (1st ed.). Westport, Connecticut: Greenwood Press.

Noh, S. J., Tachikawa, Y., Shiiba, M., & Kim, S. (2011). Applying sequential Monte Carlo methods into a distributed hydrologic model: Lagged particle filtering approach with regularization. *Hydrology and Earth System Sciences*. https://doi.org/10.5194/hess-15-

3237-2011

Oude Essink, G. H. P. (2001). Salt water intrusion in a three-dimensional groundwater system in the Netherlands: A numerical study. *Transport in Porous Media*. https://doi.org/10.1023/A:1010625913251

Panday, S., Langevin, C. D., Niswonger, R. G., Ibaraki, M., & Hughes, J. D. (2013). MODFLOW – USG Version 1: An Unstructured Grid Version of MODFLOW for Simulating Groundwater Flow and Tightly Coupled Processes Using a Control Volume Finite-Difference Formulation. *U.S. Geological Survey*, (Techniques and Methods 6-A45).

Pathiraja, S., Anghileri, D., Burlando, P., Sharma, A., Marshall, L., & Moradkhani, H. (2018). Time-varying parameter models for catchments with land use change: The importance of model structure. *Hydrology and Earth System Sciences*. https://doi.org/10.5194/hess-22-2903-2018

Pedersen, M. S., Baxter, B., Templeton, B., Rishøj, C., Theobald, D. L., Hoegh-rasmussen, E., et al. (2008). The Matrix Cookbook. *Matrix*. https://doi.org/10.1111/j.1365-294X.2006.03161.x

Pielke, R. A. (2013). *Mesoscale Meteorological Modeling*. *Mesoscale Meteorological Modeling*. https://doi.org/10.1016/C2009-0-02981-X

Poeter, E., & Townsend, P. (1994). Assessment of Critical Flow Path for Improved Remediation Management. *Groundwater*. https://doi.org/10.1111/j.1745-6584.1994.tb00661.x

Powers, J. P., Corwin, A. B., Schmall, P. C., & Kaeck, W. E. (2007). *Construction Dewatering and Groundwater Control: New Methods and Applications*. *Construction Dewatering and Groundwater Control: New Methods and Applications*. https://doi.org/10.1002/9780470168103

Pulido, M., van Leeuwen, P. J., & Posselt, D. J. (2019). *Kernel embedded nonlinear observational mappings in the variational mapping particle filter*. Retrieved from https://arxiv.org/abs/1901.10426

RamaRao, B. S., LaVenue, A. M., De Marsily, G., & Marietta, M. G. (1995). Pilot Point Methodology for Automated Calibration of an Ensemble of conditionally Simulated Transmissivity Fields: 1. Theory and Computational Experiments. *Water Resources Research*. https://doi.org/10.1029/94WR02258

Ramgraber, M., Albert, C., & Schirmer, M. (2019). Data Assimilation and Online Parameter Optimization in Groundwater Modeling Using Nested Particle Filters. *Water Resources Research*. https://doi.org/10.1029/2018WR024408

Ramgraber, M., Camporese, M., Renard, P., Salandin, P., & Schirmer, M. (2020). Quasi-online groundwater model optimization under constraints of geological consistency based on iterative importance sampling. *Water Resources Research*. https://doi.org/10.1029/2019wr026777

Reichle, R. H., McLaughlin, D. B., & Entekhabi, D. (2002). Hydrologic Data Assimilation with the Ensemble Kalman Filter. *Monthly Weather Review*. https://doi.org/10.1175/1520-0493(2002)130<0103:hdawte>2.0.co;2

Reilly, T. E., & Harbaugh, A. W. (2004). Guidelines for Evaluating Ground-Water Flow Models. *Scientific Investigations Report 2004-5038*. https://doi.org/10.1017/CBO9781107415324.004

Renard, P. (2007). Stochastic hydrogeology: What professionals really need? *Ground Water*. https://doi.org/10.1111/j.1745-6584.2007.00340.x

Robinson, T. P., & Metternicht, G. (2006). Testing the performance of spatial interpolation techniques for mapping soil properties. *Computers and Electronics in Agriculture*. https://doi.org/10.1016/j.compag.2005.07.003

Rubin, Y., & Hubbard, S. S. (2006). *Hydrogeophysics* (Vol. 50). Springer Science & Business Media.

Ruiz, J. J., Pulido, M., & Miyoshi, T. (2013). Estimating model parameters with ensemble-based data assimilation: A review. *Journal of the Meteorological Society of Japan*. https://doi.org/10.2151/jmsj.2013-201

Sanchez-Vila, X., & Fernàndez-Garcia, D. (2016). Debates—Stochastic subsurface hydrology from theory to practice: Why stochastic modeling has not yet permeated into practitioners? *Water Resources Research*. https://doi.org/10.1002/2016WR019302

Schillings, C., & Stuart, A. M. (2017). Analysis of the Ensemble Kalman Filter for Inverse Problems. *SIAM J. Numer. Anal.*, *55*(3), 1264--1290.

Schirmer, M., Leschik, S., & Musolff, A. (2013). Current research in urban hydrogeology - A review. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2012.06.015

Schölkopf, B., & Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schöniger, A., Nowak, W., & Hendricks Franssen, H. J. (2012). Parameter estimation by ensemble Kalman filters with transformed data: Approach and application to hydraulic tomography. *Water Resources Research*, *48*(4). https://doi.org/10.1029/2011WR010462

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference on* -. https://doi.org/10.1145/800186.810616

Sivanandam, S. N., & Deepa, S. N. (2008). *Introduction to genetic algorithms*. *Introduction to Genetic Algorithms*. https://doi.org/10.1007/978-3-540-73190-0

Smith, T. J., & Marshall, L. A. (2008). Bayesian methods in hydrologic modeling: A study of recent advancements in Markov chain Monte Carlo techniques. *Water Resources Research*. https://doi.org/10.1029/2007wr006705

Snyder, C., Bengtsson, T., Bickel, P., & Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*. https://doi.org/10.1175/2008MWR2529.1

Snyder, C., Bengtsson, T., & Morzfeld, M. (2015). Performance Bounds for Particle Filters Using the Optimal Proposal. *Monthly Weather Review*. https://doi.org/10.1175/mwr-d-15-0144.1

Spantini, A., Bigona, D., & Marzouk, Y. (2018). Inference via low-dimensional couplings. *The Journal of Machine Learning Research*, *19*(1), 2639–2709.

Spantini, A., Baptista, R., & Marzouk, Y. (2019). *Coupling techniques for nonlinear ensemble filtering*. Retrieved from https://arxiv.org/abs/1907.00389

Spearman, C. (1987). The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*. https://doi.org/10.2307/1422689

Stein, C. (1972). A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. Retrieved from http://static.stevereads.com/papers_to_read/a_bound_for_the_error_in_the_normal_approximation_to_the_distribution_of_a_sum_of_dependent_random_variables.pdf

Stein, C., Diaconis, P., Holmes, S., & Reinert, G. (2004). Use of exchangeable pairs in the analysis of simulations. https://doi.org/10.1214/lnms/1196283797

Stoyanov, S. V., Rachev, S. T., Racheva-Yotova, B., & Fabozzi, F. J. (2011). Fat-tailed models for risk estimation. *Journal of Portfolio Management*. https://doi.org/10.3905/jpm.2011.37.2.107

Straubhaar, J. (2019). *DeeSse user's guide*.

Sun, A. Y., Morris, A. P., & Mohanty, S. (2009). Sequential updating of multimodal hydrogeologic parameter fields using localization and clustering techniques. *Water Resources Research*. https://doi.org/10.1029/2008WR007443

Sun, N.-Z. (1994). *Inverse problems in groundwater modeling*. *Theory and applications of transport in porous media*.

Tang, Q., Kurtz, W., Brunner, P., Vereecken, H., & Hendricks Franssen, H. J. (2015). Characterisation of river-aquifer exchange fluxes: The role of spatial patterns of riverbed hydraulic conductivities. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2015.08.019

Tang, Q., Kurtz, W., Schilling, O. S., Brunner, P., Vereecken, H., & Hendricks Franssen, H. J. (2017). The influence of riverbed heterogeneity patterns on river-aquifer exchange fluxes under different connection regimes. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2017.09.031

Tjelmeland, H., & Hegstad, B. K. (2001). Mode jumping proposals in MCMC. *Scandinavian Journal of Statistics*. https://doi.org/10.1111/1467-9469.00232

Townsend, A. A. R. (2003). Genetic Algorithms - A Tutorial. Retrieved from https://pdfs.semanticscholar.org/eccb/f6523d2d29a5f6dbed9d7a0210e5ded49b96.pdf

Trippi, R. R., & DeSieno, D. (1992). Trading Equity Index Futures With a Neural Network. *The Journal of Portfolio Management*. https://doi.org/10.3905/jpm.1992.409432

UNICEF. (2016). *Professional Water Well Drilling: A UNICEF Guidance Note*. Retrieved from http://skat.ch/wp-content/uploads/2017/04/UNICEF_GuidanceNote_ProfessionalWellDrilling.pdf

Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*.

Vögeli, E. (2018). *In Fehraltorf wird die Luppmen zur Kempt*. Fehraltorf. Retrieved from https://www.fehraltorf.ch/wAssets/docs/gemeinde/geschichte_chronik/verschiedenes_zur_dorfgeschichte/Luppmen-Kempt.pdf

Vrugt, J. A., ter Braak, C. J. F., Diks, C. G. H., & Schoups, G. (2013). Hydrologic data assimilation using particle Markov chain Monte Carlo simulation: Theory, concepts and applications. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2012.04.002

Wendt, J. F., Anderson, J. D., Degroote, J., Degrez, G., Dick, E., Grundmann, R., & Vierendeels, J. (2009). *Computational fluid dynamics: An introduction*. *Computational Fluid Dynamics*. https://doi.org/10.1007/978-3-540-85056-4

Werner, A. D., Bakker, M., Post, V. E. A., Vandenbohede, A., Lu, C., Ataie-Ashtiani, B., et al. (2013). Seawater intrusion processes, investigation and management: Recent advances and future challenges. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2012.03.004

Werner, D. (2018). *Funktionalanalysis* (8th ed.). Berlin: Springer Spektrum. https://doi.org/https://doi.org/10.1007/978-3-662-55407-4

White, J. T. (2018). A model-independent iterative ensemble smoother for efficient history-matching and uncertainty quantification in very high dimensions. *Environmental Modelling and Software*. https://doi.org/10.1016/j.envsoft.2018.06.009

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*. https://doi.org/10.1016/0169-7439(87)80084-9

Yan, H., Moradkhani, H., & Zarekarizi, M. (2017). A probabilistic drought forecasting framework: A combined dynamical and statistical approach. *Journal of Hydrology*. https://doi.org/10.1016/j.jhydrol.2017.03.004

Yan, H., Zarekarizi, M., & Moradkhani, H. (2018). Toward improving drought monitoring using the remotely sensed soil moisture assimilation: A parallel particle filtering framework. *Remote Sensing of Environment*. https://doi.org/10.1016/j.rse.2018.07.017

Yeh, W. W. (1986). Review of Parameter Identification Procedures in Groundwater Hydrology: The Inverse Problem. *Water Resources Research*. https://doi.org/10.1029/WR022i002p00095

Zhou, H., Gómez-Hernández, J. J., Hendricks Franssen, H. J., & Li, L. (2011). An approach to handling non-Gaussianity of parameters and state variables in ensemble Kalman filtering. *Advances in Water Resources*, *34*(7), 844–864. https://doi.org/10.1016/j.advwatres.2011.04.014

Zhu, G., Li, X., Ma, J., Wang, Y., Liu, S., Huang, C., et al. (2018). A new moving strategy for the sequential Monte Carlo approach in optimizing the hydrological model parameters. *Advances in Water Resources*. https://doi.org/10.1016/j.advwatres.2018.02.007

Zovi, F. (2014). *Assessment of heterogeneous hydraulic properties in natural*

aquifers at the intermediate scale. *PhD thesis*. University of Padova. Retrieved from http://paduaresearch.cab.unipd.it/6708/

Zovi, F., Camporese, M., Hendricks Franssen, H. J., Huisman, J. A., & Salandin, P. (2017). Identification of high-permeability subsurface structures with multiple point geostatistics and normal score ensemble Kalman filter. *Journal of Hydrology*, *548*, 208–224. https://doi.org/10.1016/j.jhydrol.2017.02.056

I Supporting Information for Chapter 2

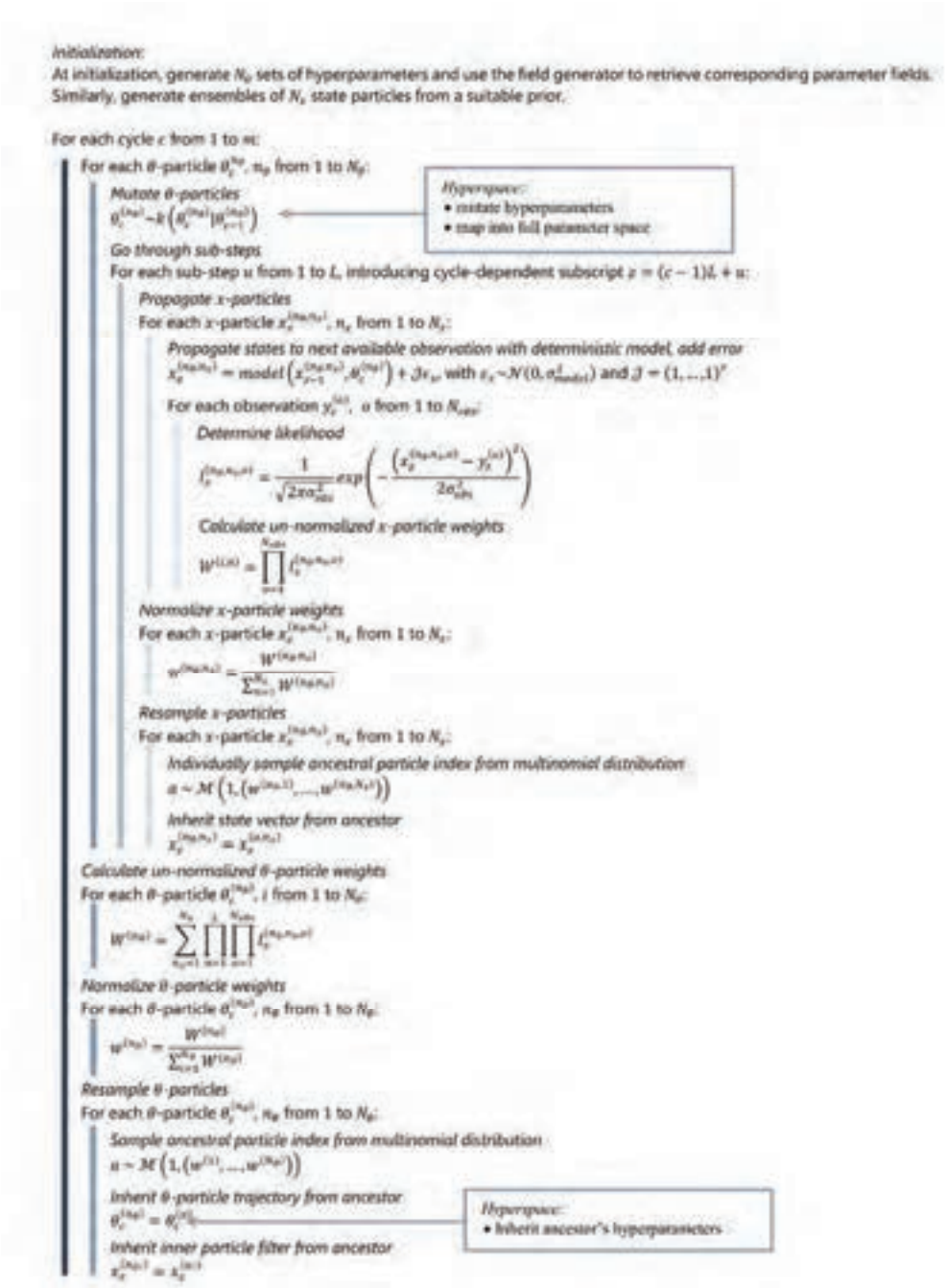


Figure I-1. Pseudo-code for the nested particle filter algorithm as employed in this study.

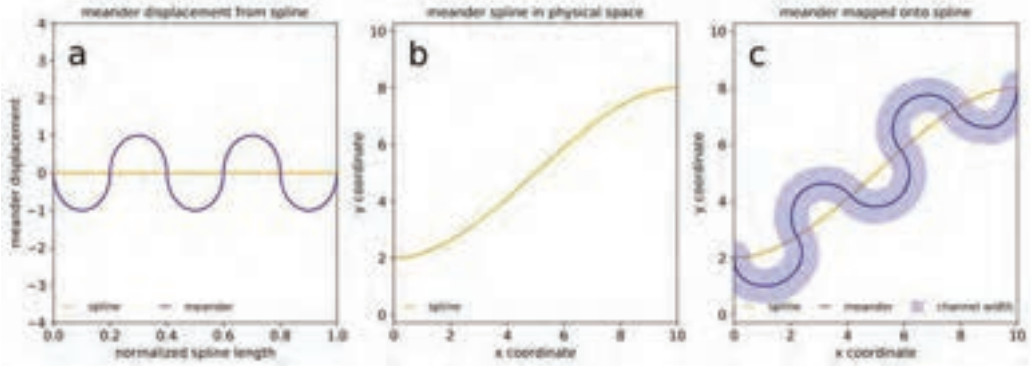


Figure I-2. Conceptual steps of the meander generator: the meander displacement relative to a parent spline is determined, based on hyperparameters 'number of meander turns' and 'phase shift' (a); the parent spline is created in physical space, based on hyperparameters 'start point', 'end point', and their 'first derivatives' (b); meander displacement is multiplied by 'meander width' and mapped onto the parent spline; adherence to 'meander facies' is determined based on hyperparameter 'channel width' (c).

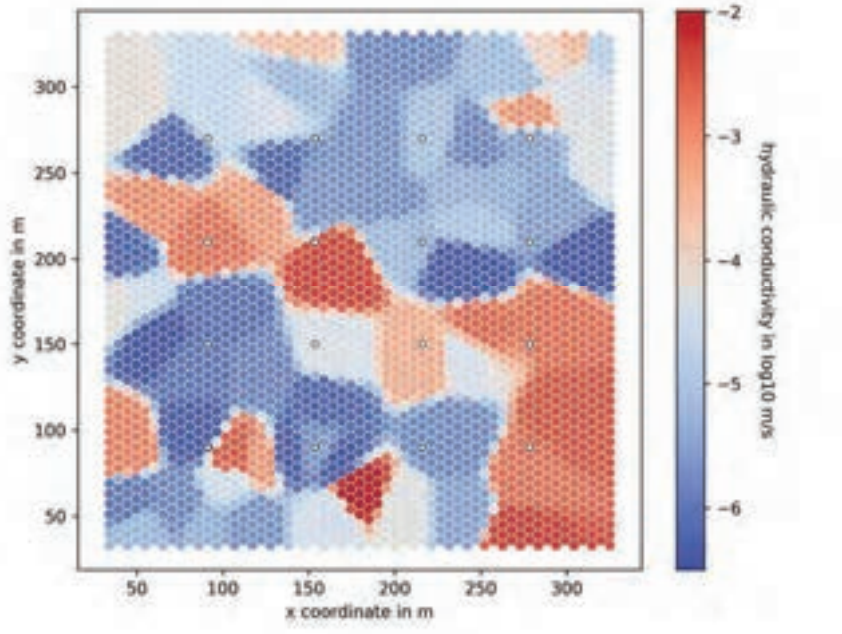


Figure I-3. Ensemble average of hydraulic conductivity for the hybrid nested particle (node-based scenario) at the end of the assimilation period. As in the nested particle filter scenarios, parameter uncertainty has collapsed.

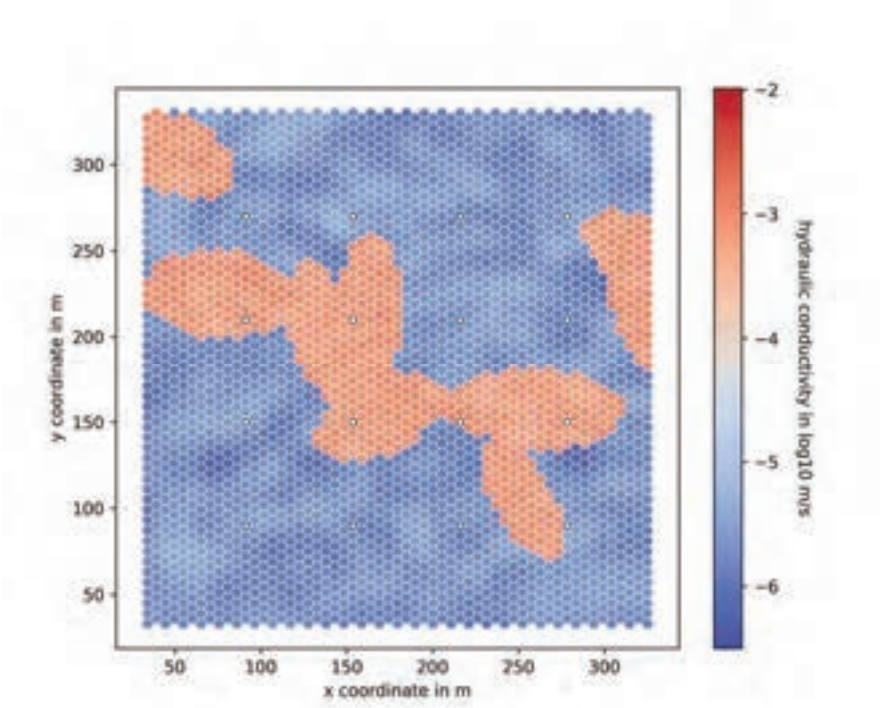


Figure I-4. Ensemble average of hydraulic conductivity for the hybrid nested particle (lens-based scenario) at the end of the assimilation period. As in the nested particle filter scenarios, parameter uncertainty has collapsed.

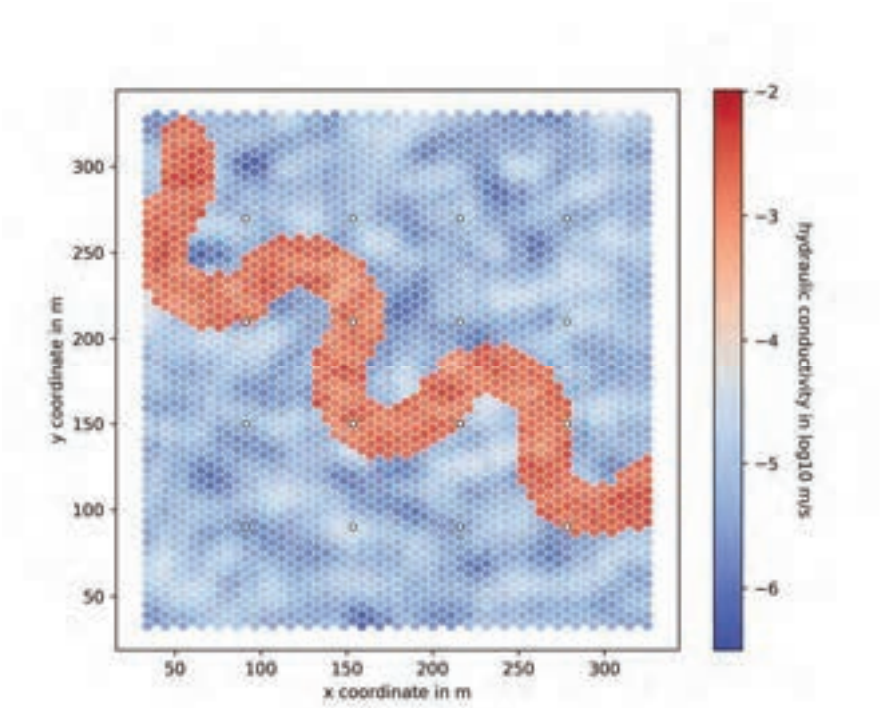


Figure I-5. Ensemble average of hydraulic conductivity for the hybrid nested particle (meander-based scenario) at the end of the assimilation period. As in the nested particle filter scenarios, parameter uncertainty has collapsed.

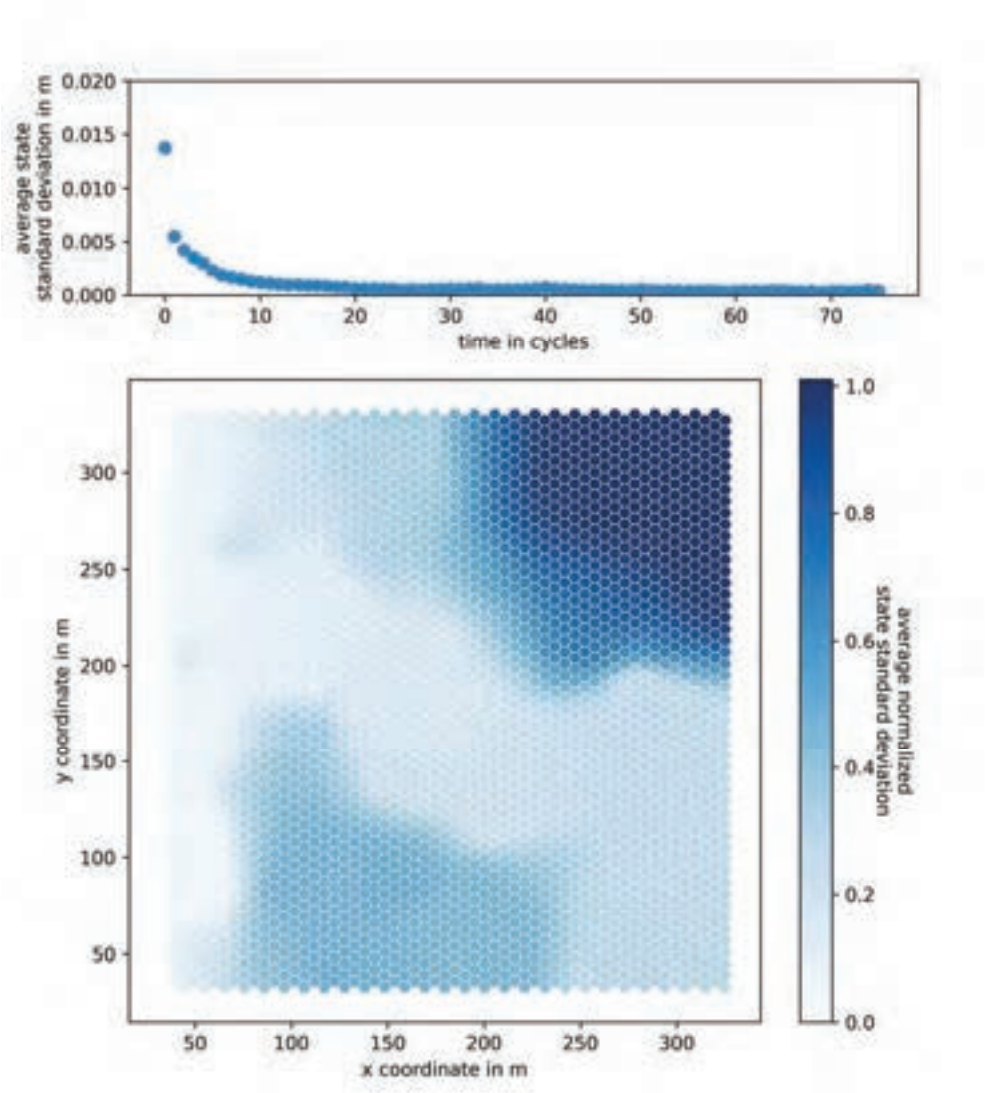


Figure I-6. Development of the standard deviation of the inner EnKFs' state uncertainties averaged across all inner filters and grid cells over time (top). The lower subplot shows the normalized average of the inner EnKFs' spatial standard deviation distribution. Results are shown for the node-based scenario of the hybrid nested particle filter.

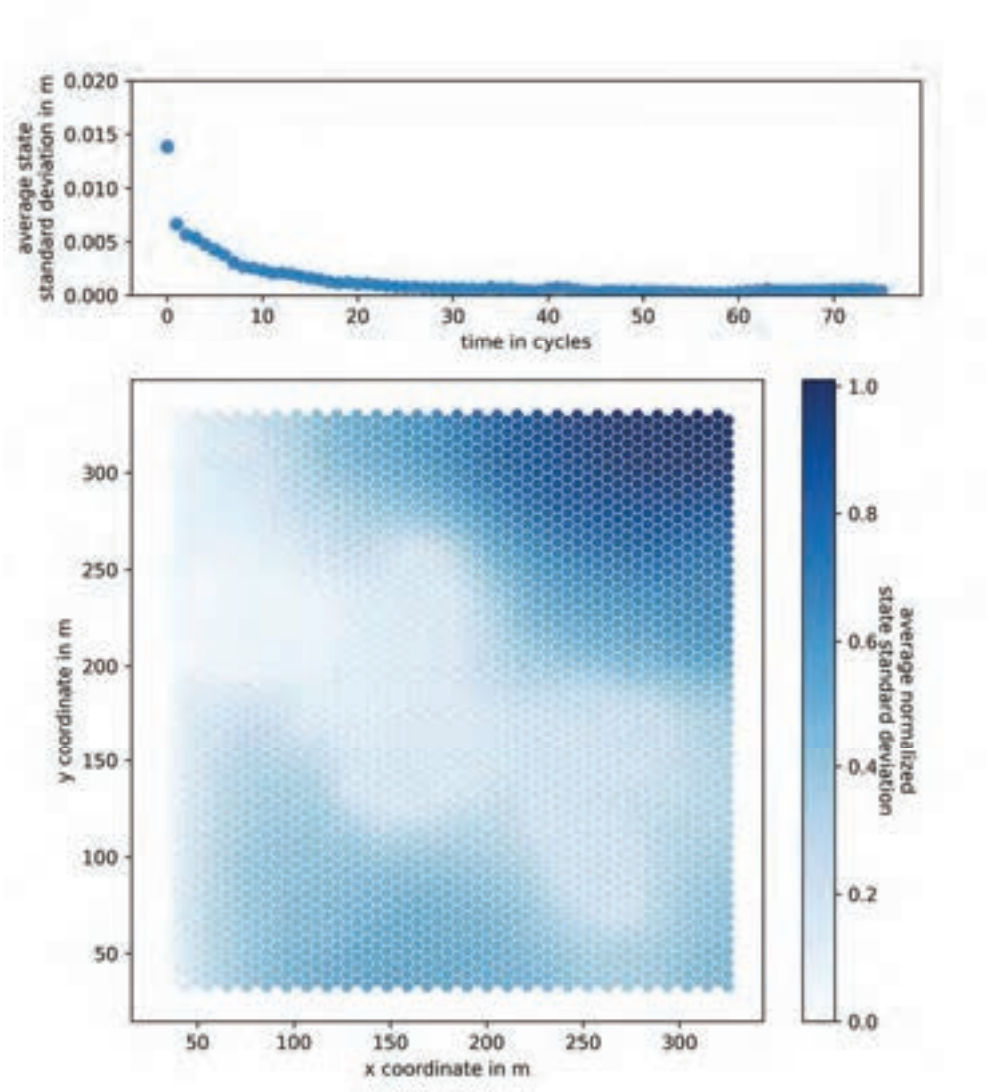


Figure I-7. Development of the standard deviation of the inner EnKFs' state uncertainties averaged across all inner filters and grid cells over time (top). The lower subplot shows the normalized average of the inner EnKFs' spatial standard deviation distribution. Results are shown for the lens-based scenario of the hybrid nested particle filter.

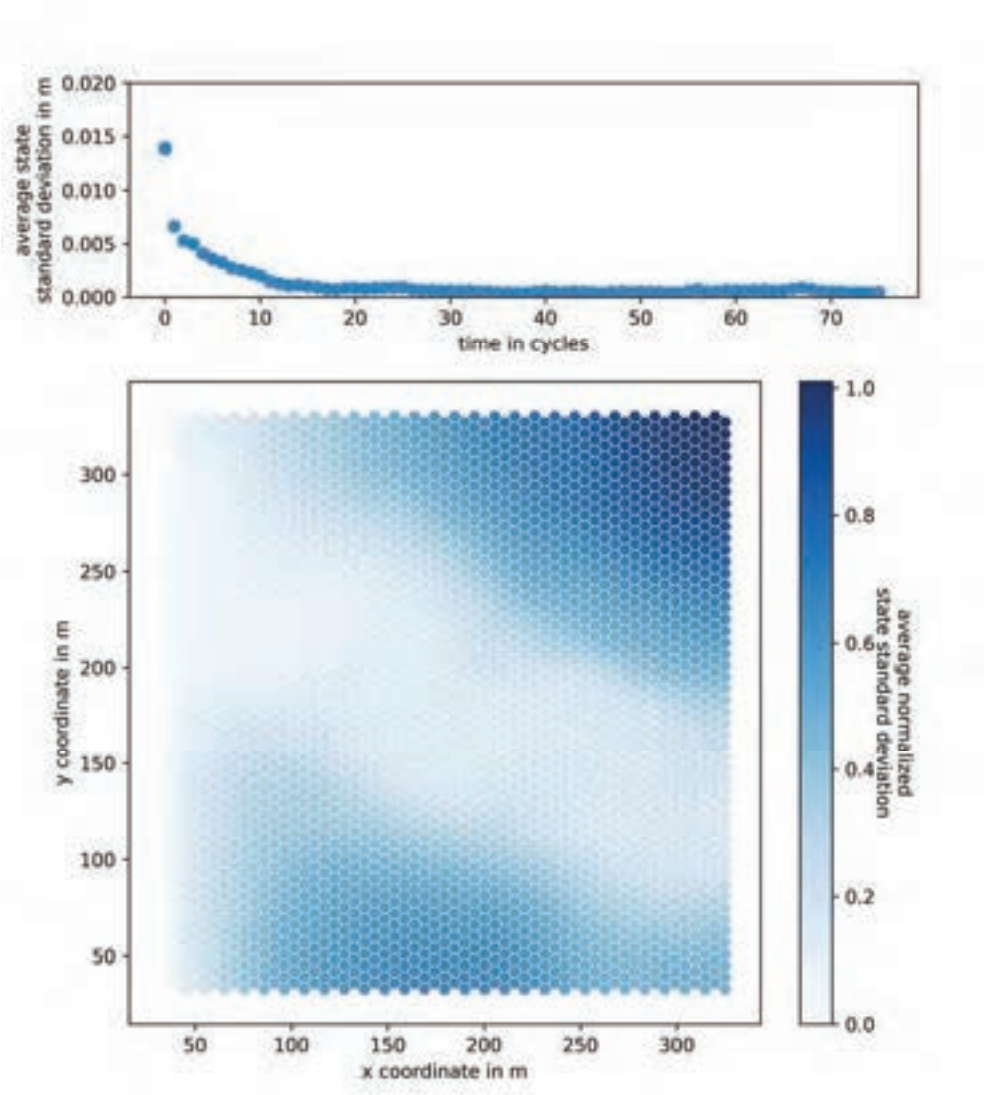


Figure I-8. Development of the standard deviation of the inner EnKFs' state uncertainties averaged across all inner filters and grid cells over time (top). The lower subplot shows the normalized average of the inner EnKFs' spatial standard deviation distribution. Results are shown for the meander-based scenario of the hybrid nested particle filter.

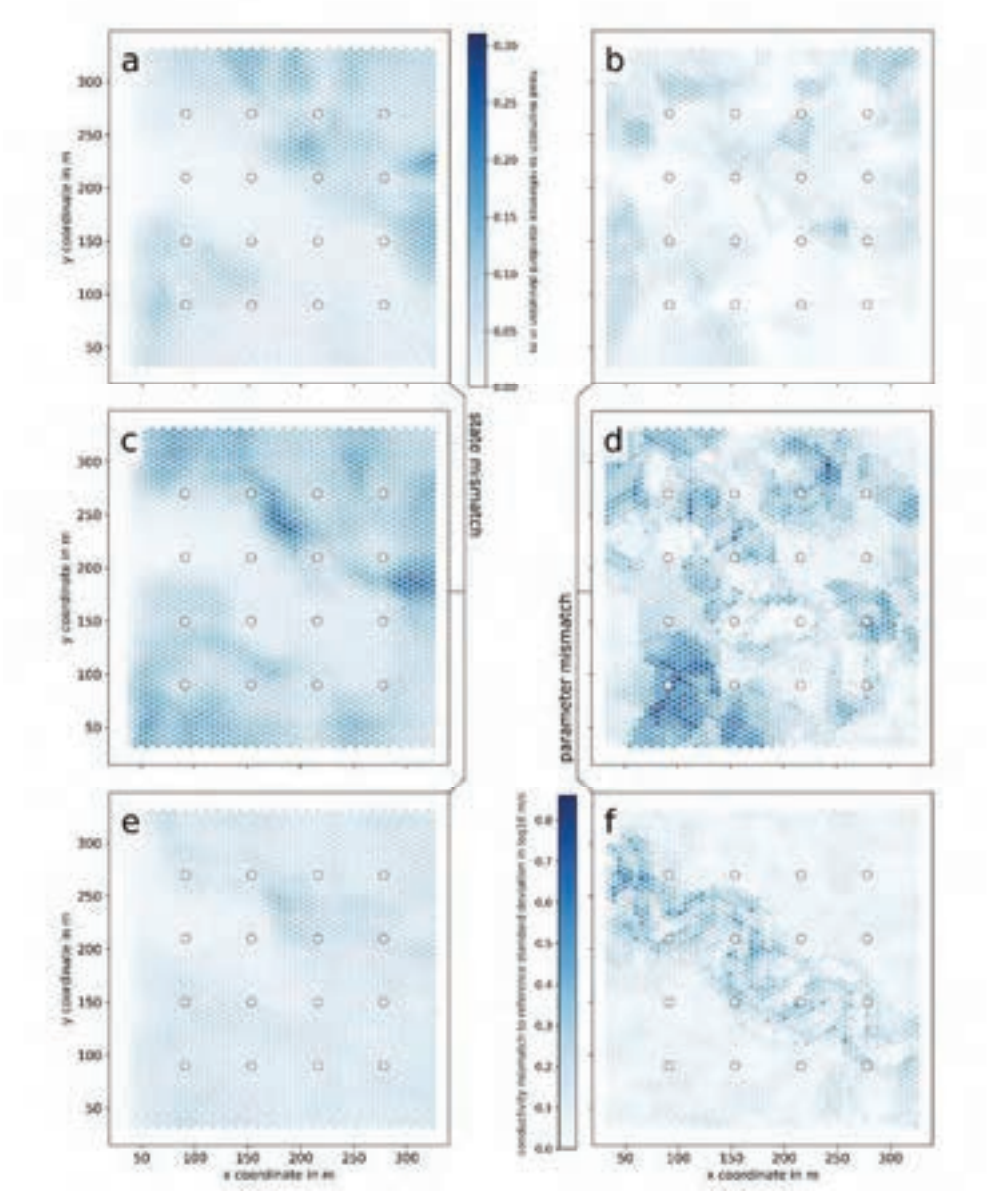


Figure I-9. Standard deviation of the mismatch between the synthetic reference and ensemble means at the end of the data assimilation period for hydraulic heads (a, c, e) and hydraulic conductivities (b, d, f) across ten random seeds. Results are shown for the node-based (a, b), lens-based (c, d), and meander-based (e, f) hyperparameterizations.

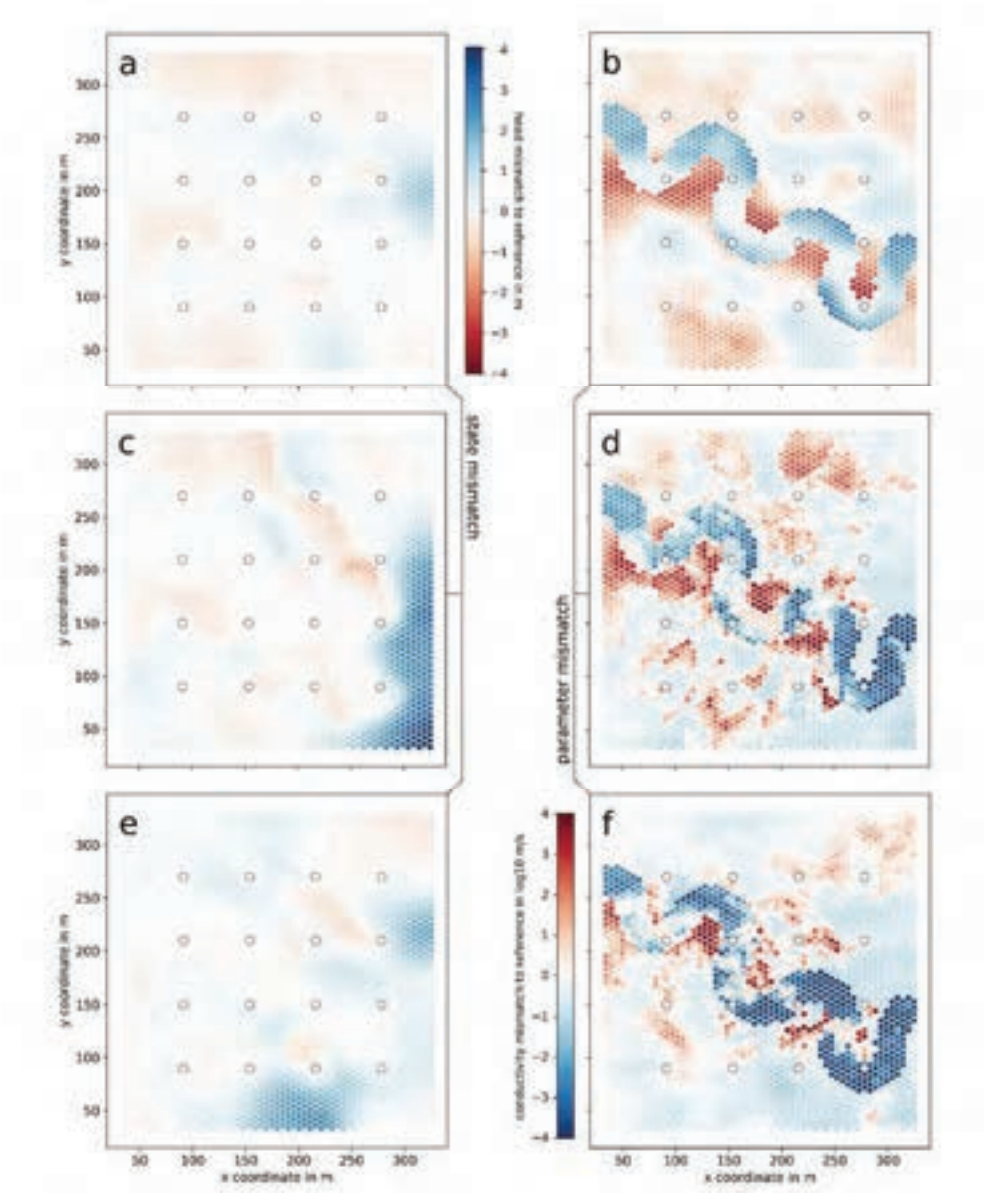


Figure I-10. Standard EnKF mismatch between the synthetic reference and ensemble mean at the end of the data assimilation period for hydraulic heads (a, c, e) and hydraulic conductivities (b, d, f). Results are shown for the node-based (a, b), lens-based (c, d), and meander-based (e, f) hyperparameterizations. Mind that the relation of colors to quantities is reversed between the state mismatch (a, c, e) and the parameter mismatch (b, d, f) columns. This was a deliberate choice to visually underline the common relation of parameter underestimation to state overestimation, and vice versa.

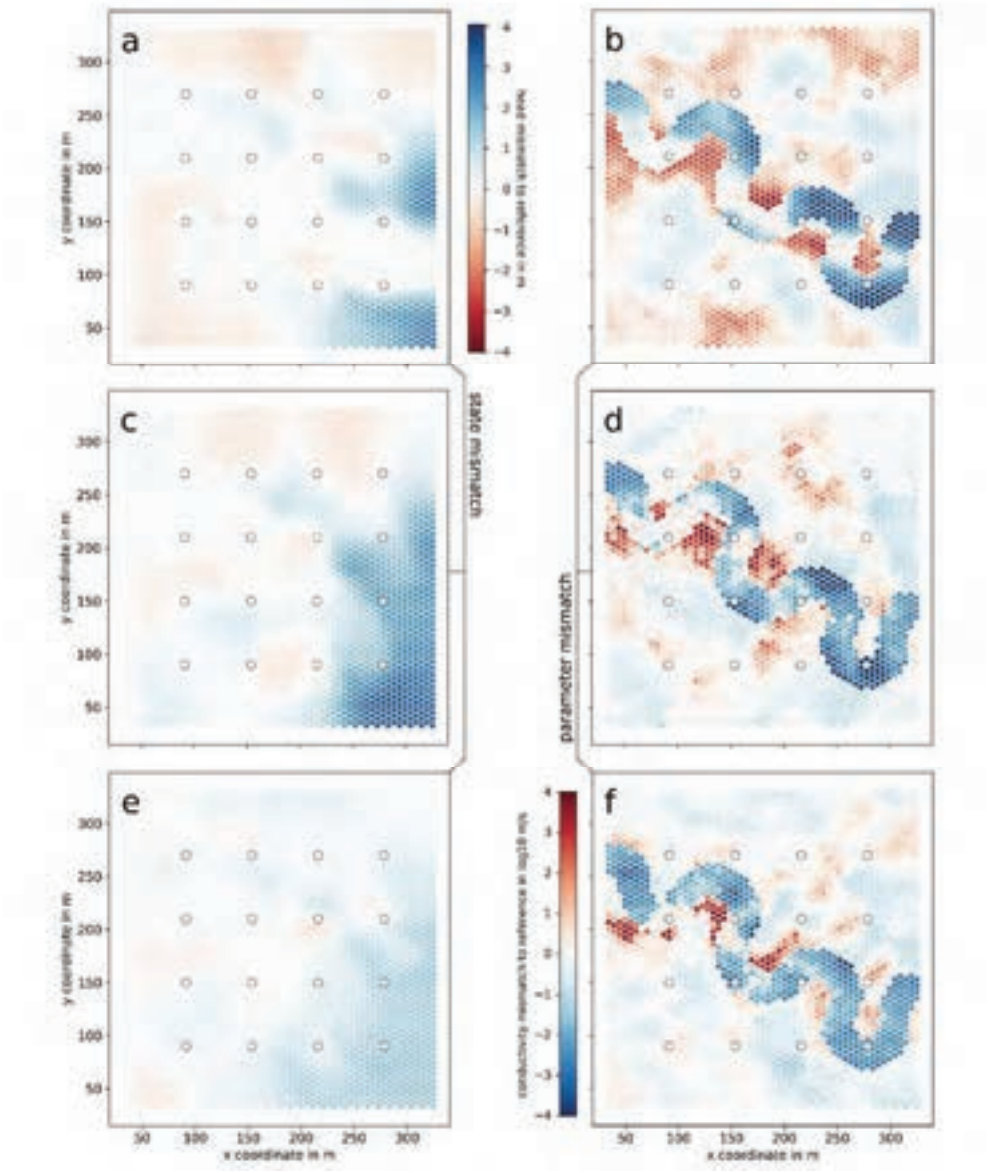


Figure I-11. GA-EnKF mismatch between the synthetic reference and ensemble mean at the end of the data assimilation period for hydraulic heads (a, c, e) and hydraulic conductivities (b, d, f). Results are shown for the node-based (a, b), lens-based (c, d), and meander-based (e, f) hyperparameterizations. Mind that the relation of colors to quantities is reversed between the state mismatch (a, c, e) and the parameter mismatch (b, d, f) columns. This was a deliberate choice to visually underline the common relation of parameter underestimation to state overestimation, and vice versa.

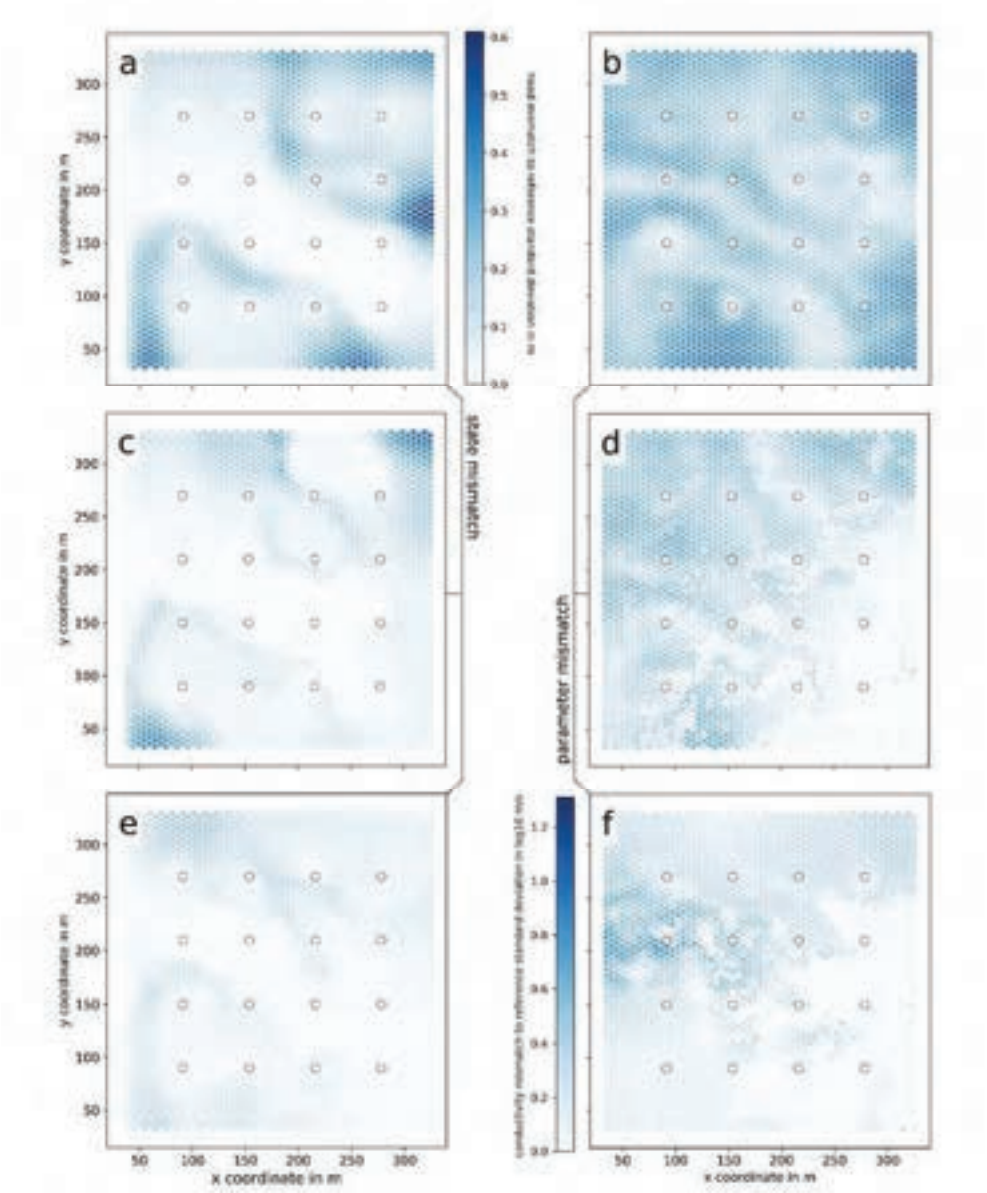


Figure I-12. Standard deviation of the mismatch between the synthetic reference and ensemble of the standard EnKF at the end of the data assimilation period for hydraulic heads (a, c, e) and hydraulic conductivities (b, d, f). Results are shown for the node-based (a, b), lens-based (c, d), and meander-based (e, f) hyperparameterizations.

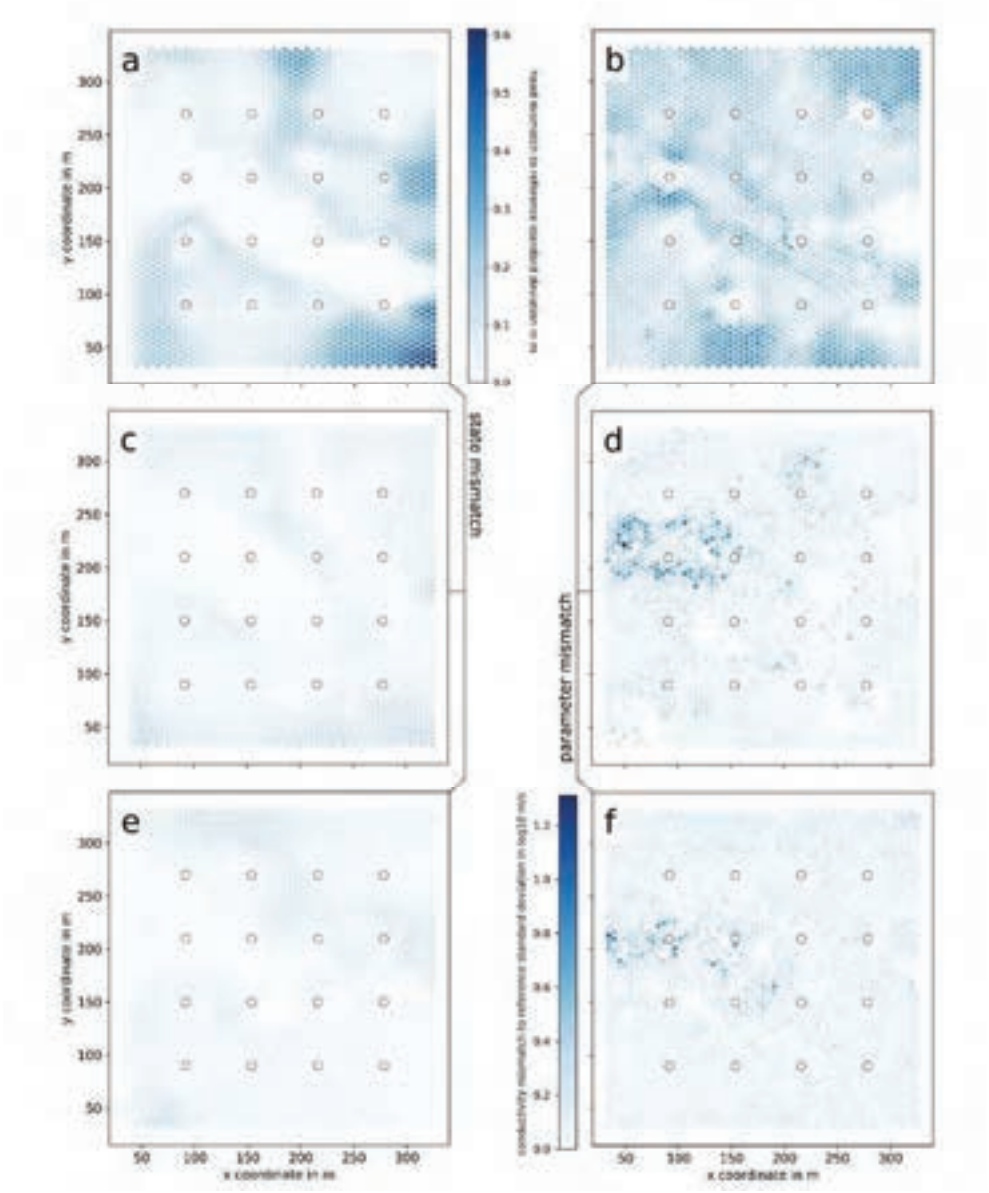


Figure I-13. Standard deviation of the mismatch between the synthetic reference and ensemble of the GA-EnKF at the end of the data assimilation period for hydraulic heads (a, c, e) and hydraulic conductivities (b, d, f). Results are shown for the node-based (a, b), lens-based (c, d), and meander-based (e, f) hyperparameterizations.

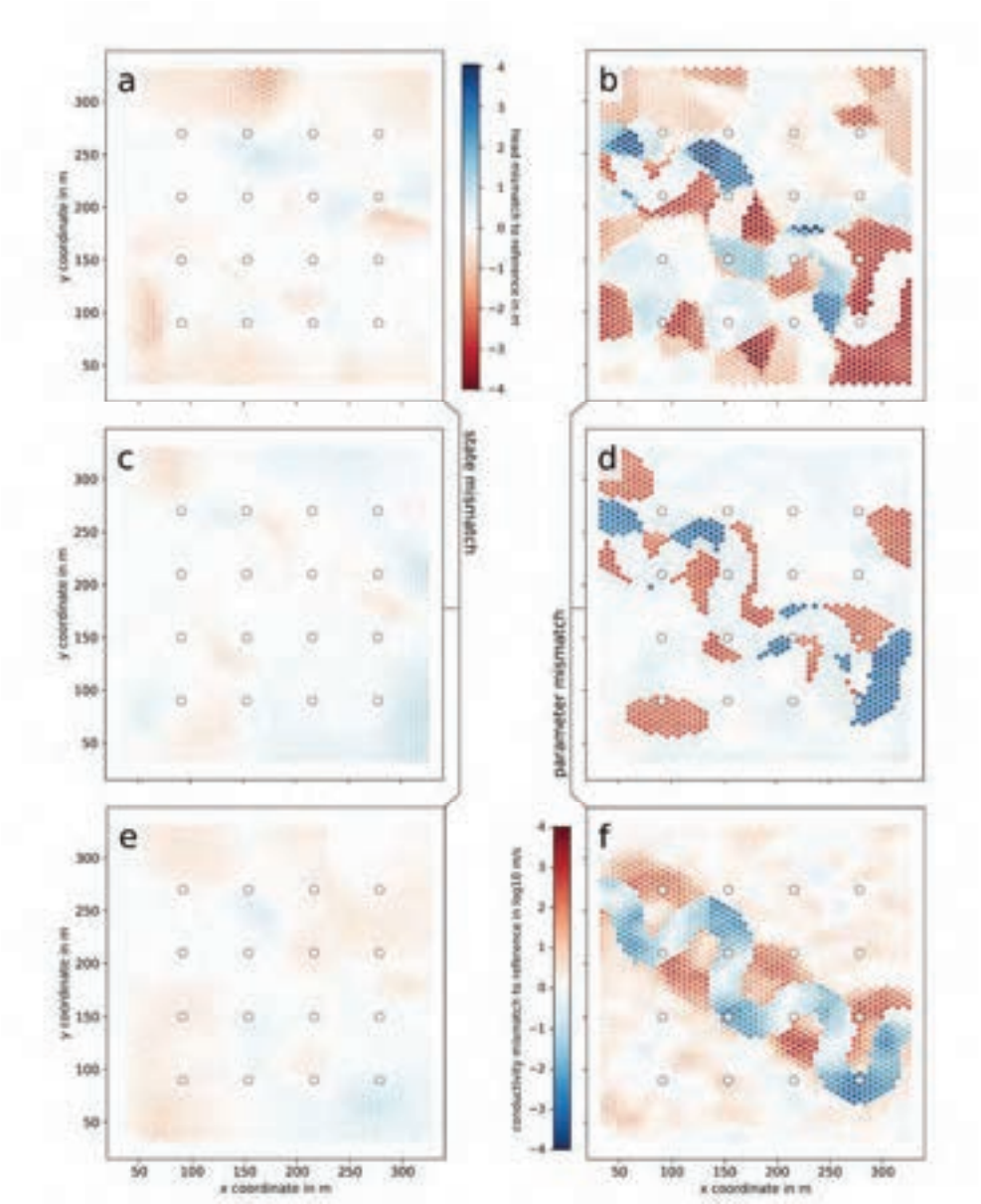


Figure I-14. Mismatch between the synthetic reference and ensemble mean at the end of the data assimilation period for hydraulic heads (a, c, e) and hydraulic conductivities (b, d, f) for the hybrid nested particle filter with inner EnKFs. Results are shown for the node-based (a, b), lens-based (c, d), and meander-based (e, f) hyperparameterizations. Mind that the relation of colors to quantities is reversed between the state mismatch (a, c, e) and the parameter mismatch (b, d, f) columns. This was a deliberate choice to visually underline the common relation of parameter underestimation to state overestimation, and vice versa.

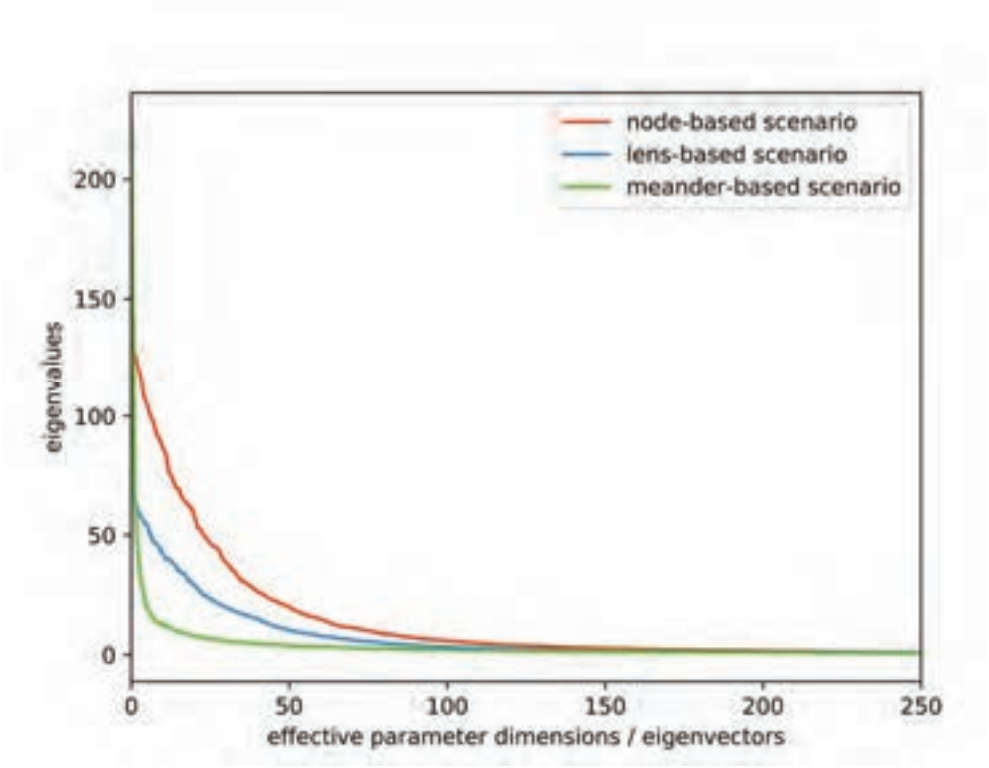


Figure I-15. Eigenvalues of the covariance matrices of the initial parameter ensembles for the three different geological characterizations. The node-based case has the largest effective dimensionality (the most non-zero eigenvalues), followed by the lens-based scenario. The meander-based scenario, perhaps unsurprisingly, has the fewest effective dimensions.

Table I-1. Variables used in model discretization and filter setup.

Model variable	Value
Cell count	2850
Hexagon radius	3 m
Cell top elevation	10 m
Cell bottom elevation	-10 m
Recharge period length	25 d
Recharge mean	1.07E-9 m/s
Total simulation time	750 d
Filter variable	**Value**
State measurement frequency	every 24 h
N_θ	200
N_x	5 per parameter particle
L	10
σ_M	0.0001 m
σ_{obs}	0.02 m

Table I-2. Random mutation operations for the node-based field generator.

Chance	Random operation
10%	Add or remove a random node, chance dependent on whether node count is below or above the target count of 50 nodes. New nodes are placed randomly and assigned a random hydraulic conductivity.
5%	Remove a random node and add a new one.
15%	Move a random node within a user-specified radius, here 50 m.
35%	Change a random node's log hydraulic conductivity by adding a random value drawn from a standard normal distribution. Log hydraulic conductivities are bounded between -2.5 and -6.
15%	Adopt hydraulic conductivity of a random node within a radius of 100 m.
20%	Switch hydraulic conductivity of two random nodes within a radius of 100 m of each other.

Table I-3. Random mutation operations for the lens-based field generator.

Chance	Random operation
10%	Add or remove a random lens, chance dependent on whether the lens count is below or above target count of 12 lenses. New lenses are placed randomly and assigned random rotation, size, and aspect.
20%	Remove a random lens and add a new one.
30%	Move a random lens within a user-specified radius, here 50 m.
5%	Change a random lens's size by adjusting the length of its primary axis, bounded between 75 m and 90 m.
15%	Change a random lens's rotation.
5%	Change a random lens's aspect between primary and secondary axis, bounded between 1.75 and 2.25.
15%	Change hydraulic conductivity of one of the facies. Draw a value from a standard normal distribution, then generate a new Gaussian parameter field with the specified mean, isotropic spatial correlation, and an amplitude of 1. Log hydraulic conductivities are bounded between -2.5 and -6.

Table I-4. Random mutation operations for the meander-based field generator.

Chance	Random operation
10%	Move start or end point of the meander direction spline.
15%	Change first derivative of start or end of the meander direction spline, bounded between -1 and 1.
15%	Adjust number of meander turns, bounded between 5 and 9.
20%	Adjust channel width, bounded between 5 and 50 % of the meander 'wavelength'.
20%	Adjust meander phase shift.
20%	Change hydraulic conductivity of one of the facies. Draw a value from a standard normal distribution, then generate a new Gaussian parameter field with the specified mean, isotropic spatial correlation, and an amplitude of 1. Log hydraulic conductivities are bounded between -2.5 and -6.

II Supporting Information for Chapter 3

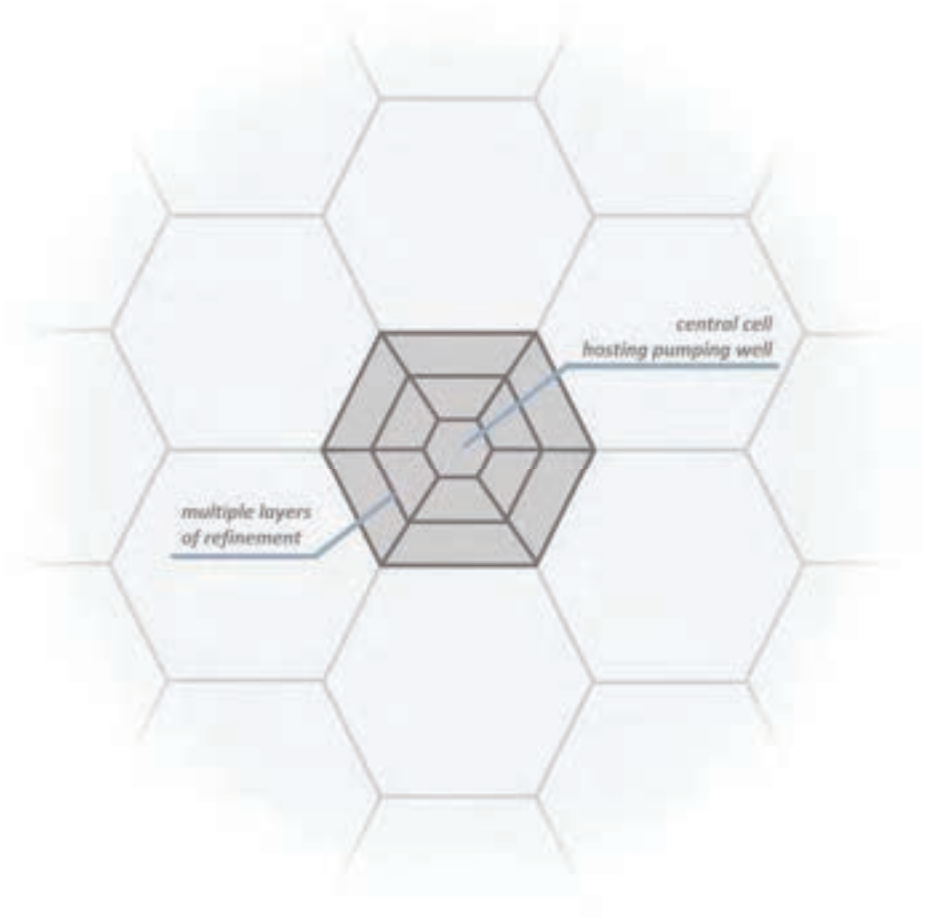


Figure II-1. Refinement of hexagonal grids around pumping wells. The web-based subdivision scheme allows for multiple layers of refinement, permitting the model to better simulate cones of depression. Here, we illustrated a two-fold refinement. In this study, we refined the cells seven times.

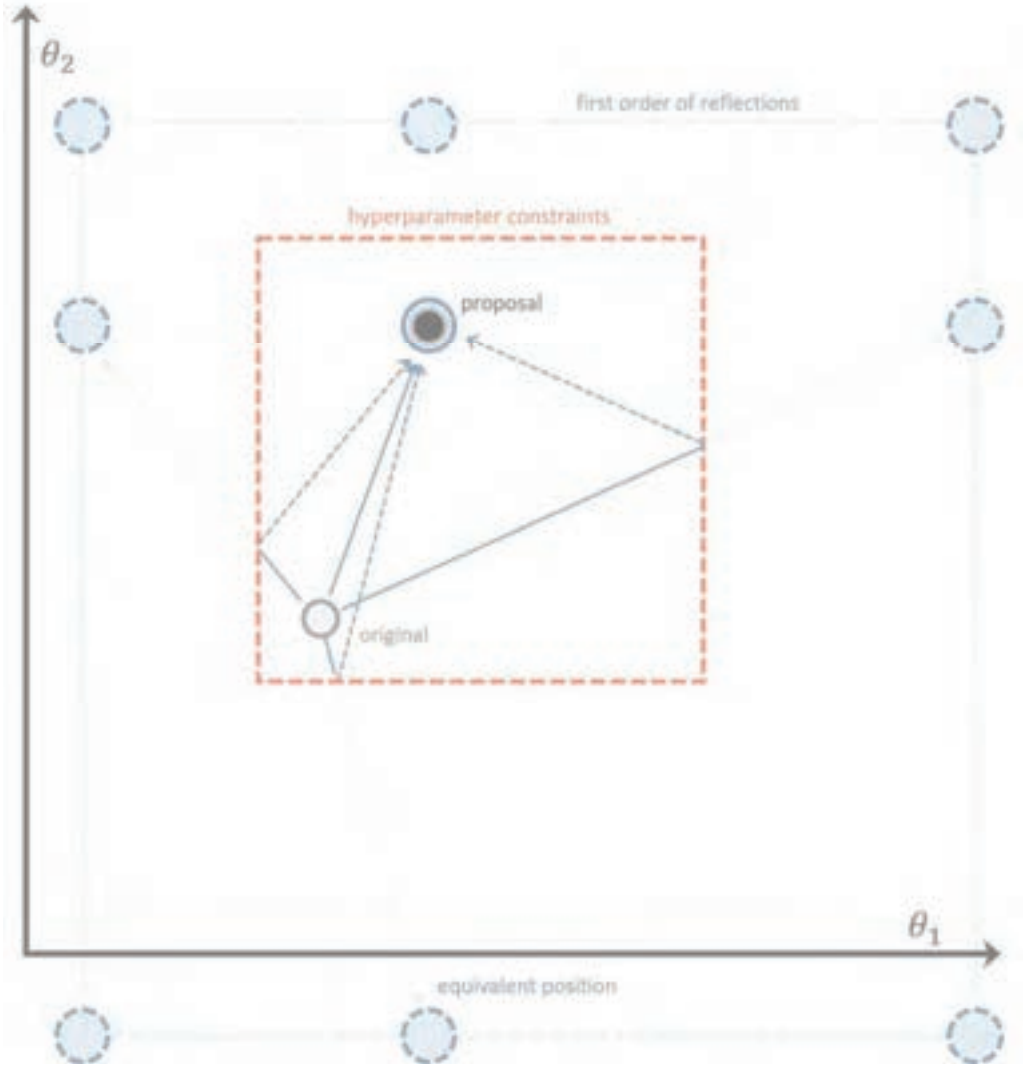


Figure II-2. Equivalent positions (dashed blue circles) to the proposal (full blue circle) along two hyperparameter space dimensions with an upper and lower limit each (red dashed rectangle). Jumps to any of the equivalent positions result in the same proposal after reflection over the hyperparameter constraints.

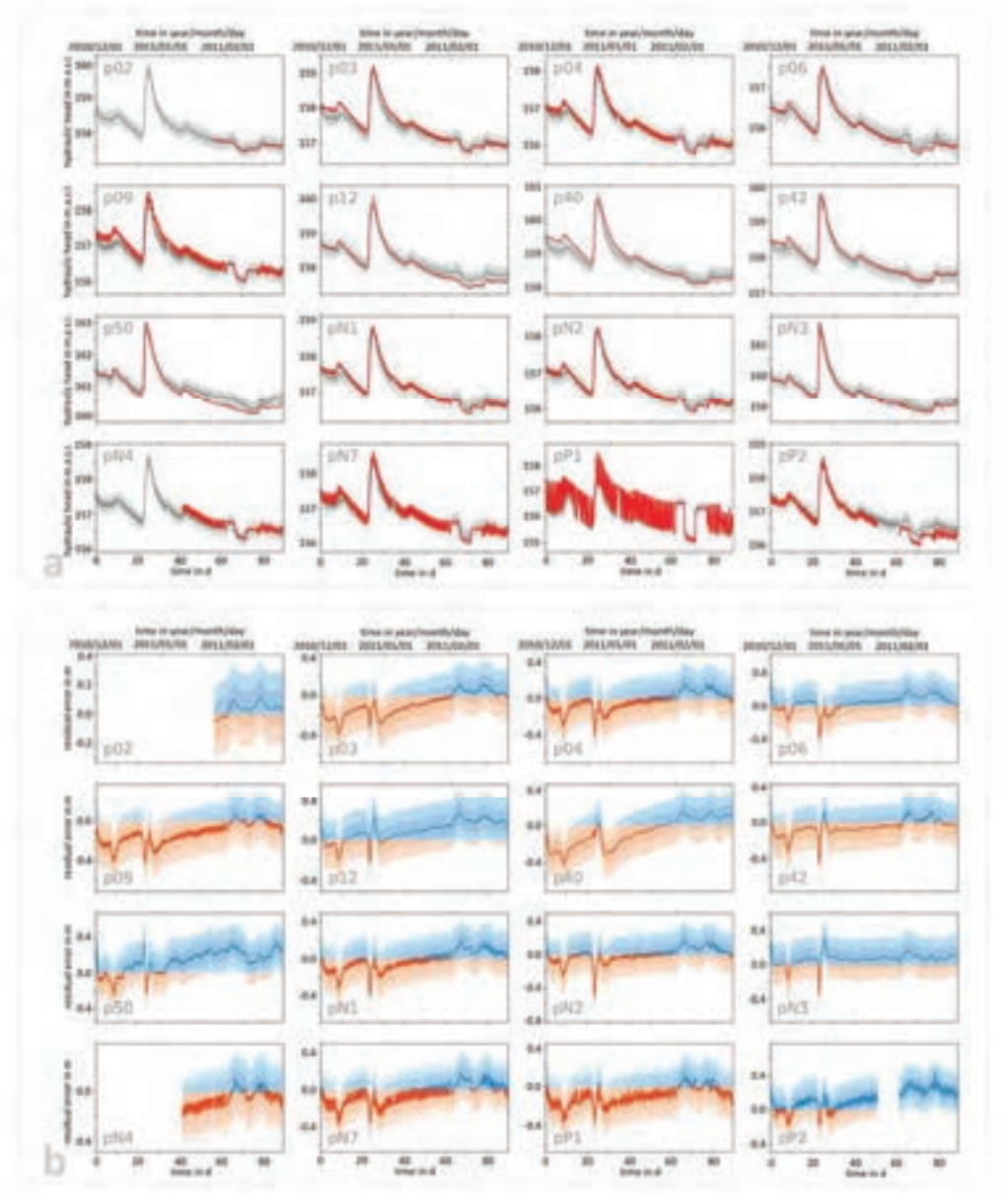


Figure II-3. Evaluation of the model predictions of Scenario 1c over the validation period for all observation and pumping wells. (a) illustrates the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation. (b) illustrates the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

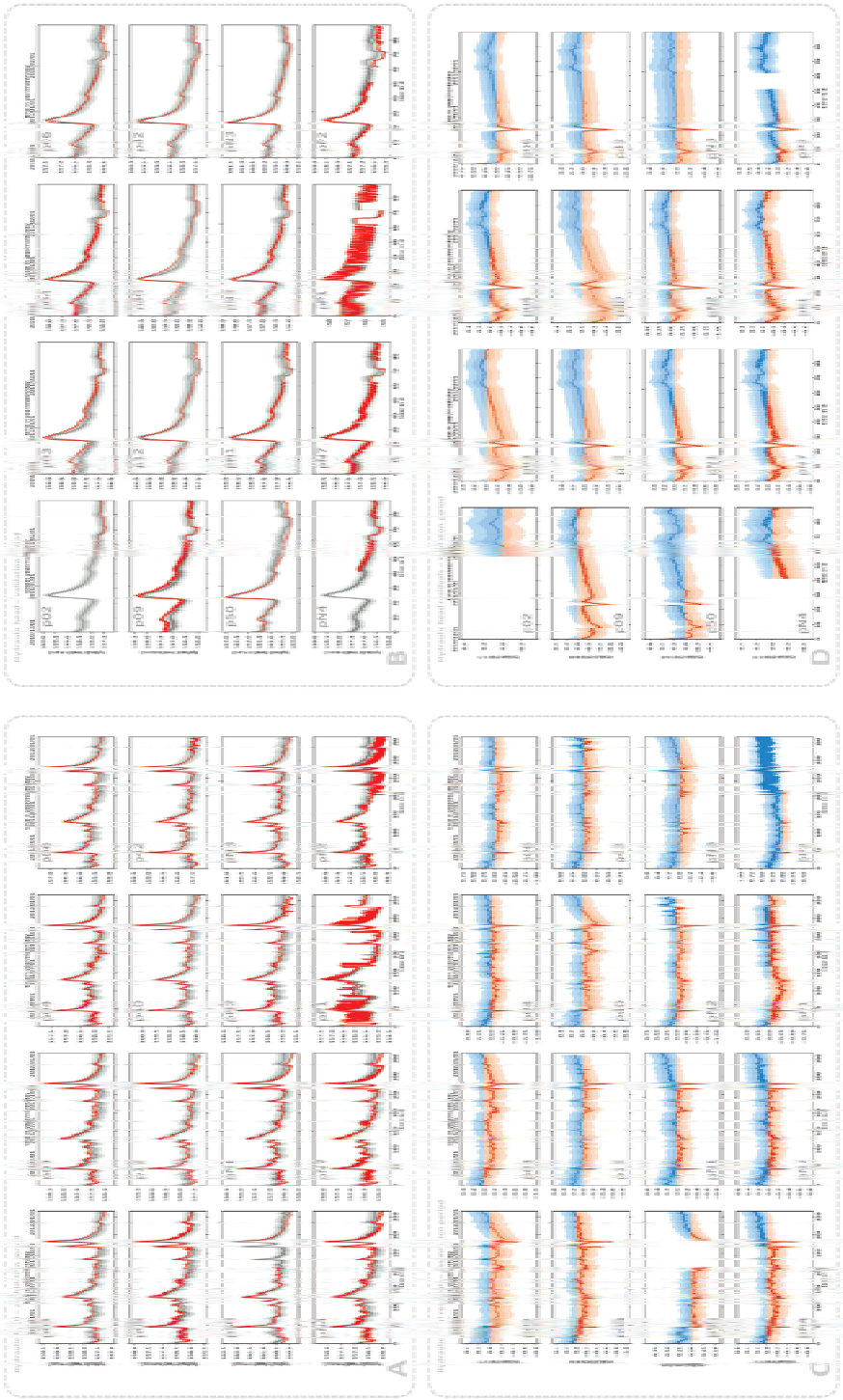


Figure II-4. Evaluation of the model predictions of Scenario 1a over the assimilation period (A, C) and validation period (B, D) for all observation and pumping wells. (A) and (B) illustrate the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation for the assimilation (A) and validation (B) periods. (C) and (D) illustrate the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

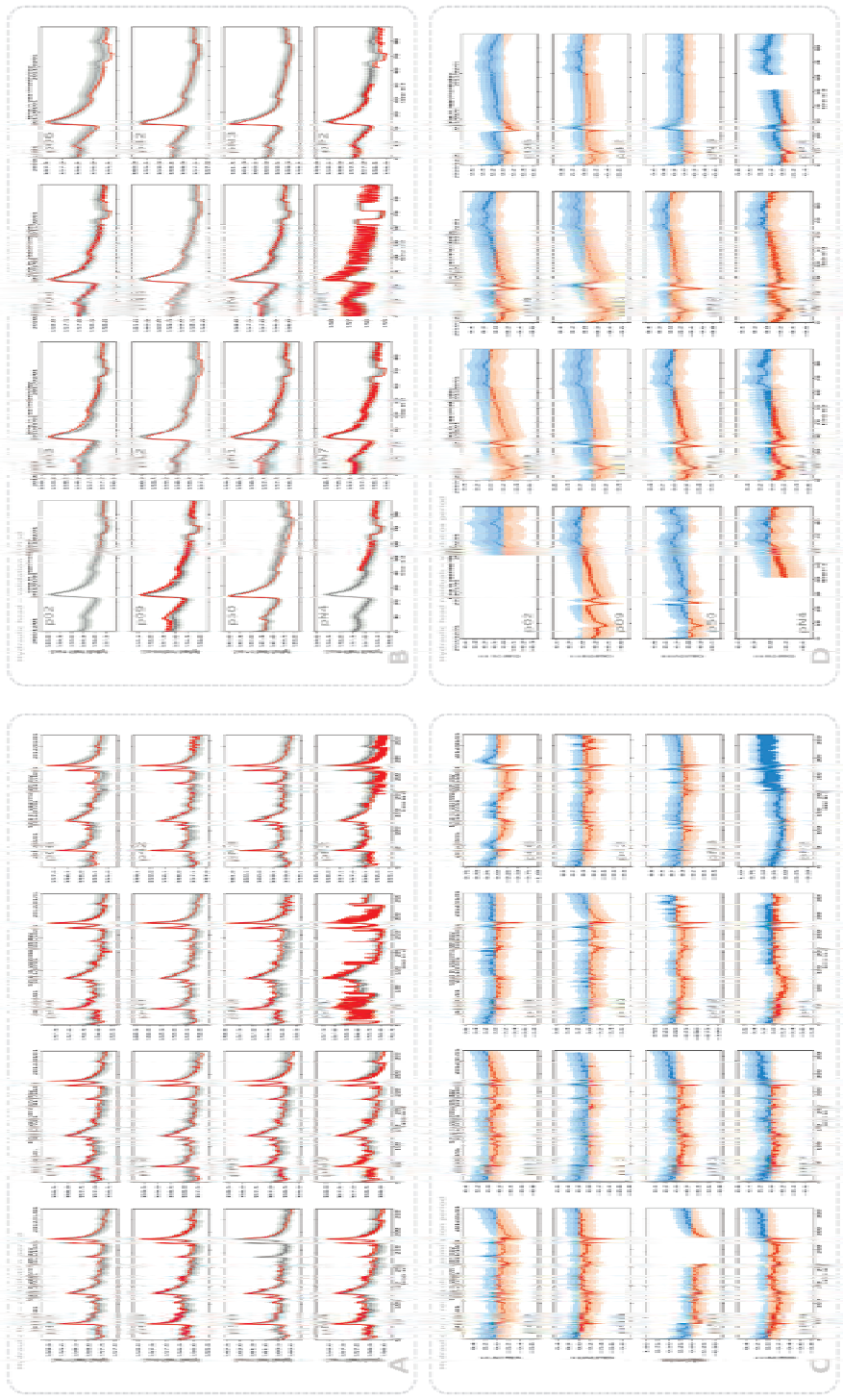


Figure II-5. Evaluation of the model predictions of Scenario 1b over the assimilation period (A, C) and validation period (B, D) for all observation and pumping wells. (A) and (B) illustrate the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation for the assimilation (A) and validation (B) periods. (C) and (D) illustrate the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

Figure II-6. Evaluation of the model predictions of Scenario 2a over the assimilation period (A, C) and validation period (B, D) for all observation and pumping wells. (A) and (B) illustrate the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation for the assimilation (A) and validation (B) periods. (C) and (D) illustrate the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

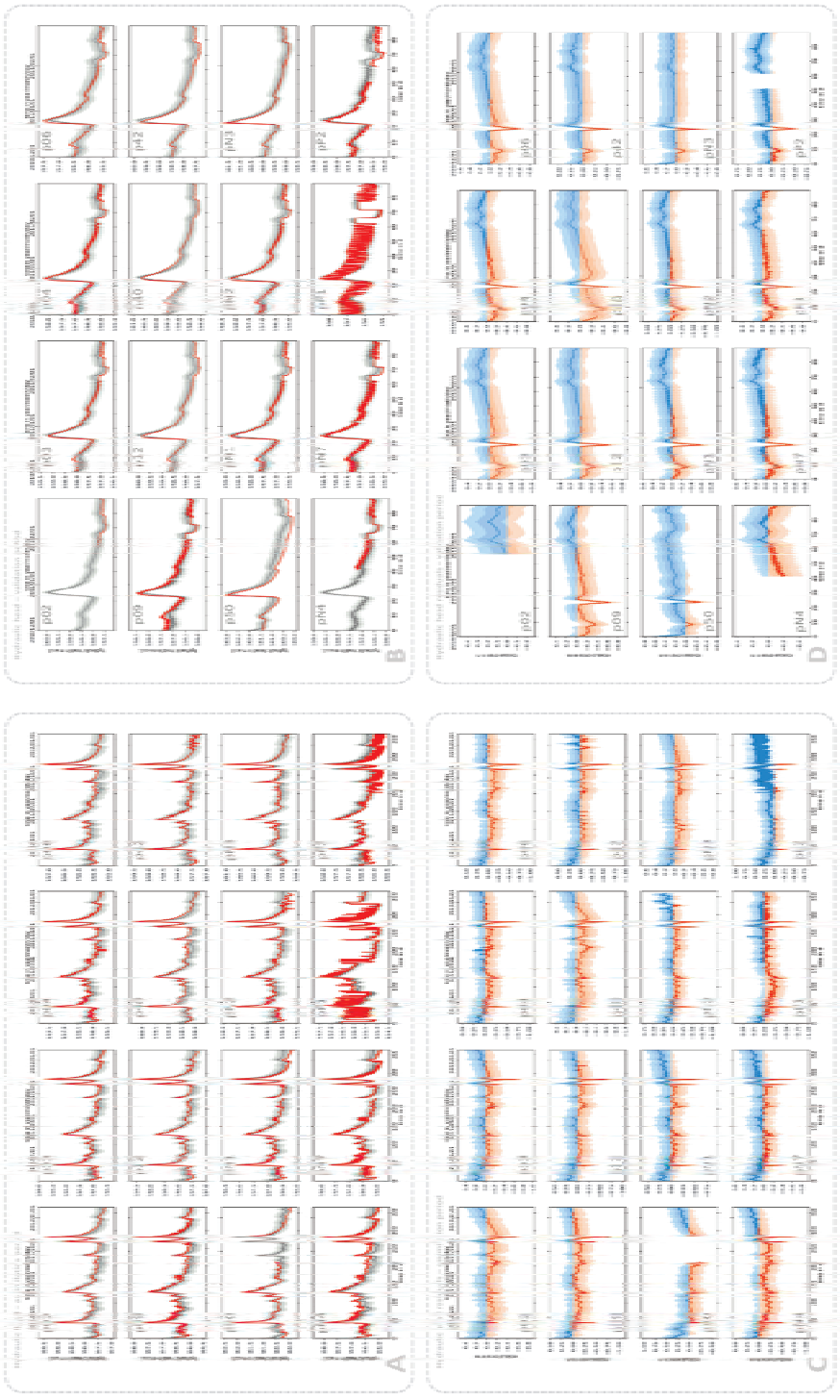


Figure II-7. Evaluation of the model predictions of Scenario 2b over the assimilation period (A, C) and validation period (B, D) for all observation and pumping wells. (A) and (B) illustrate the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation for the assimilation (A) and validation (B) periods. (C) and (D) illustrate the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

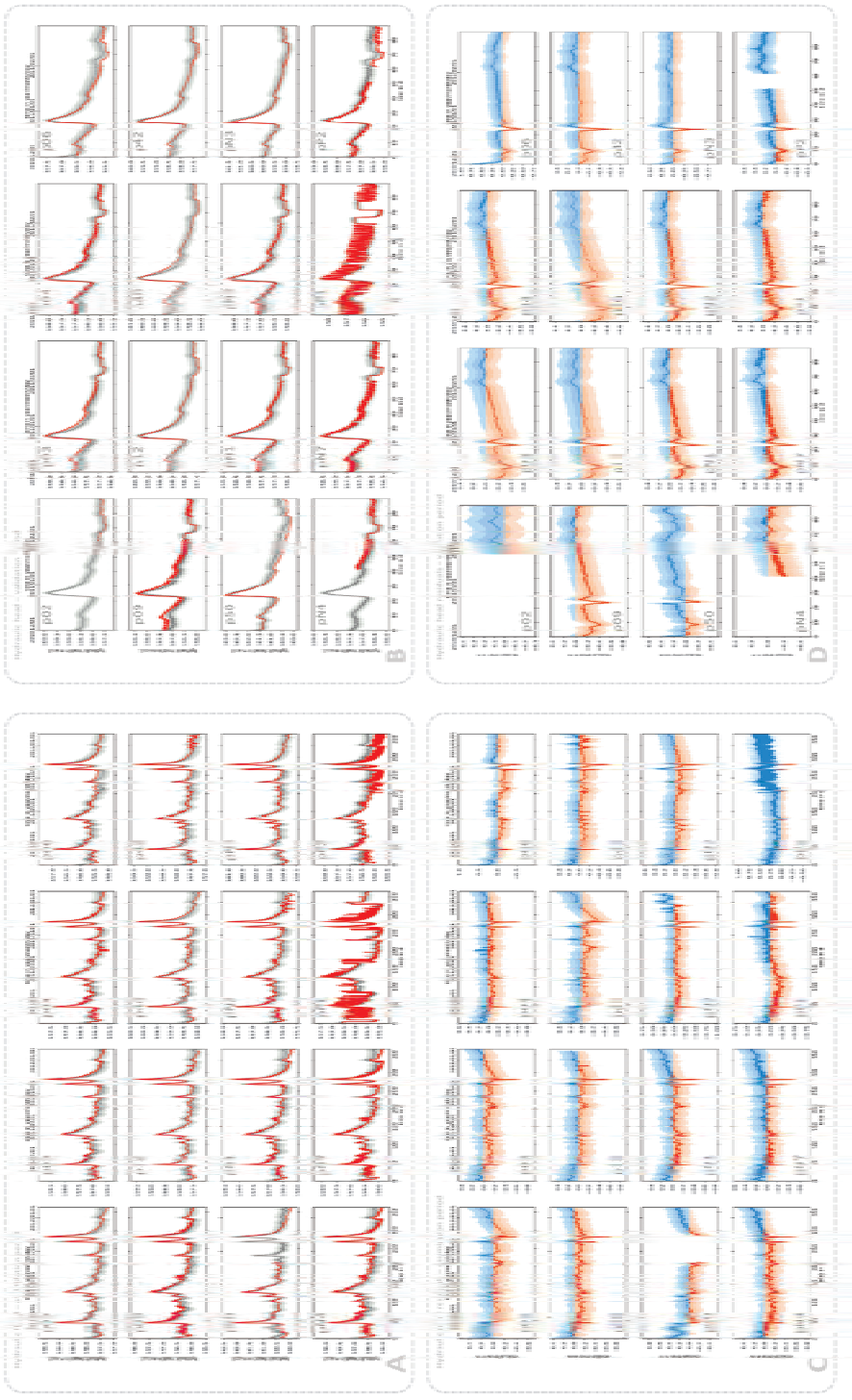


Figure II-8. Evaluation of the model predictions of Scenario 2c over the assimilation period (A, C) and validation period (B, D) for all observation and pumping wells. (A) and (B) illustrate the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation for the assimilation (A) and validation (B) periods. (C) and (D) illustrate the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

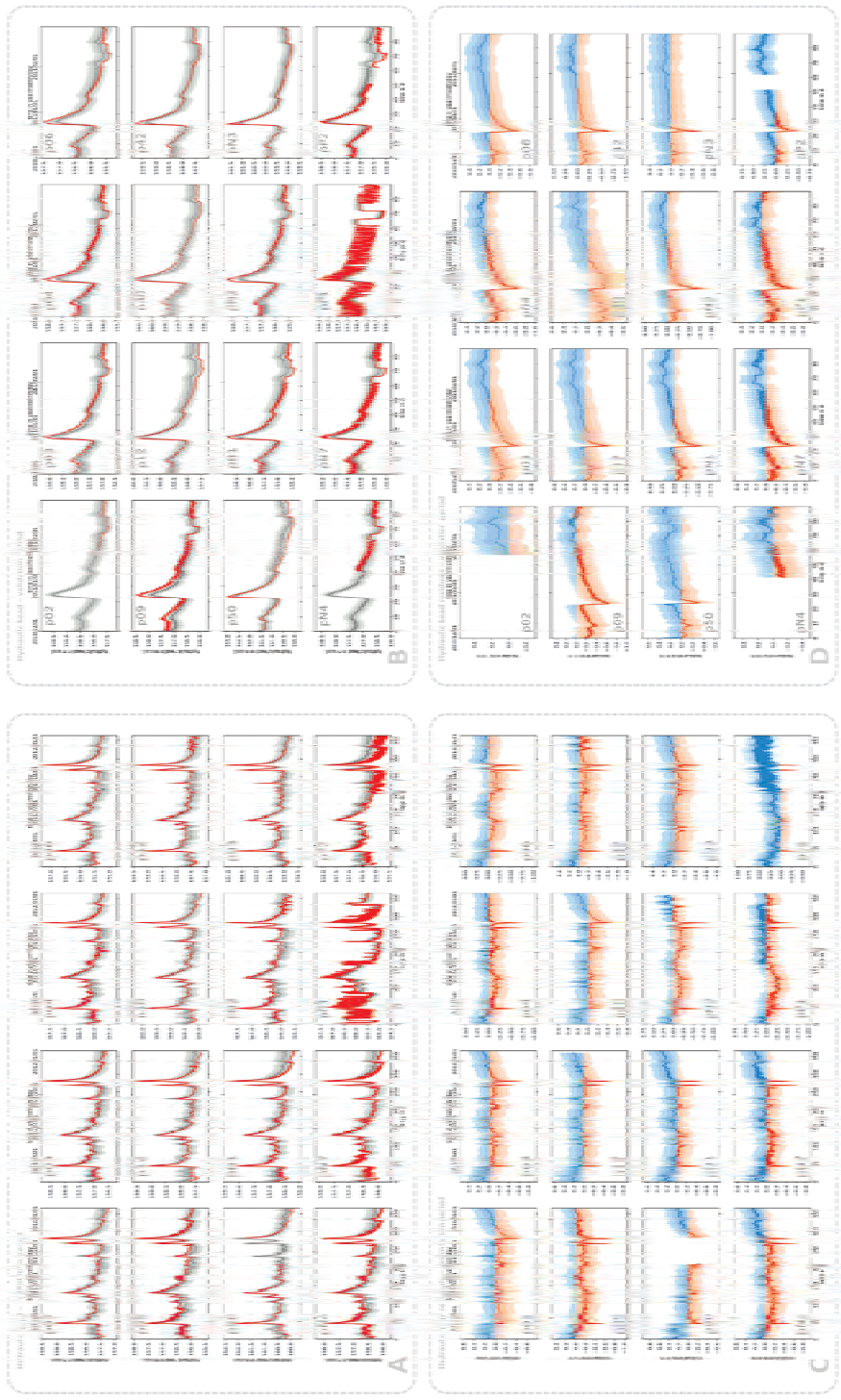


Figure II-9. Evaluation of the model predictions of Scenario 3a over the assimilation period (A, C) and validation period (B, D) for all observation and pumping wells. (A) and (B) illustrate the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation for the assimilation (A) and validation (B) periods. (C) and (D) illustrate the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

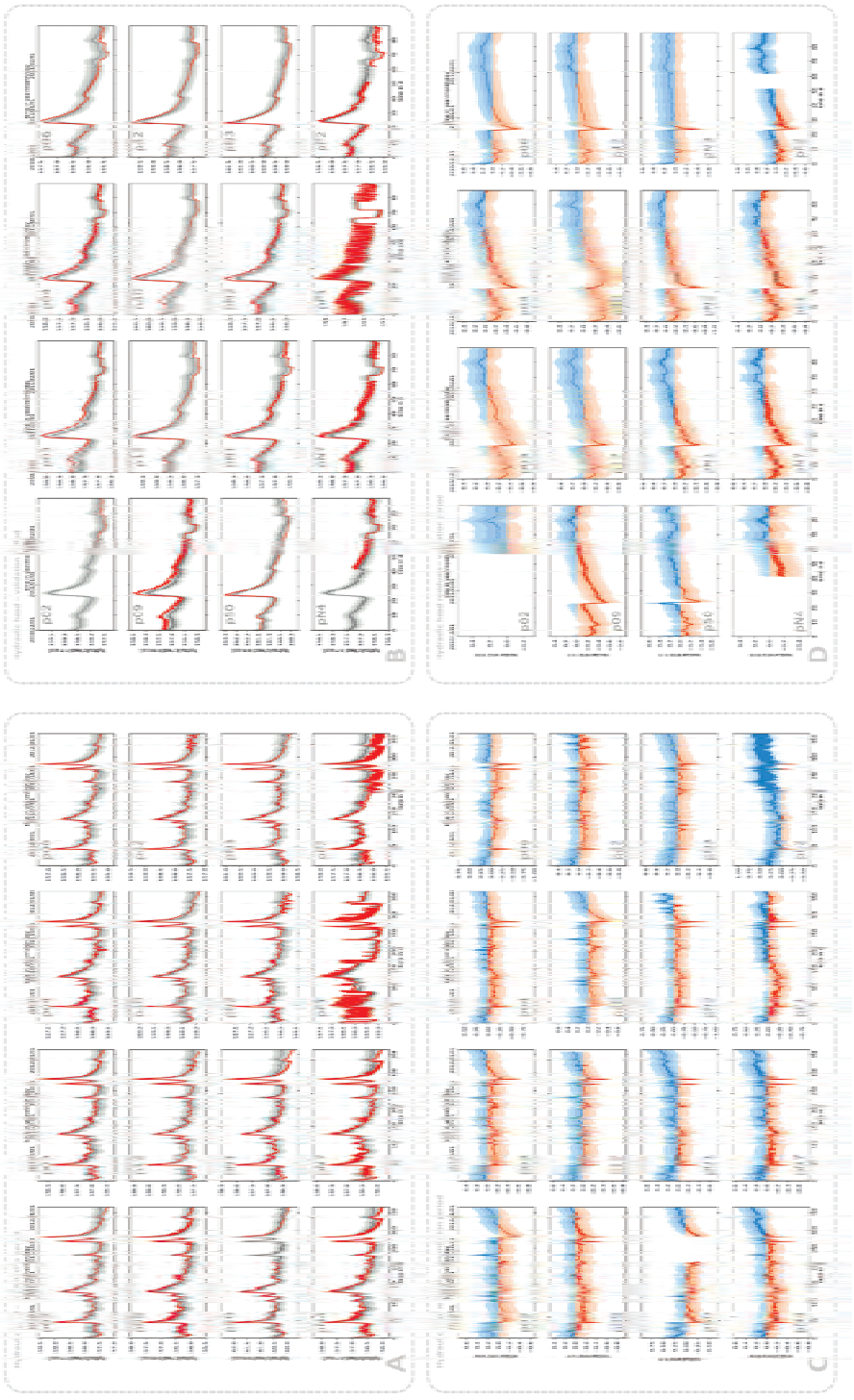


Figure II-10. Evaluation of the model predictions of Scenario 3b over the assimilation period (A, C) and validation period (B, D) for all observation and pumping wells. (A) and (B) illustrate the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation for the assimilation (A) and validation (B) periods. (C) and (D) illustrate the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

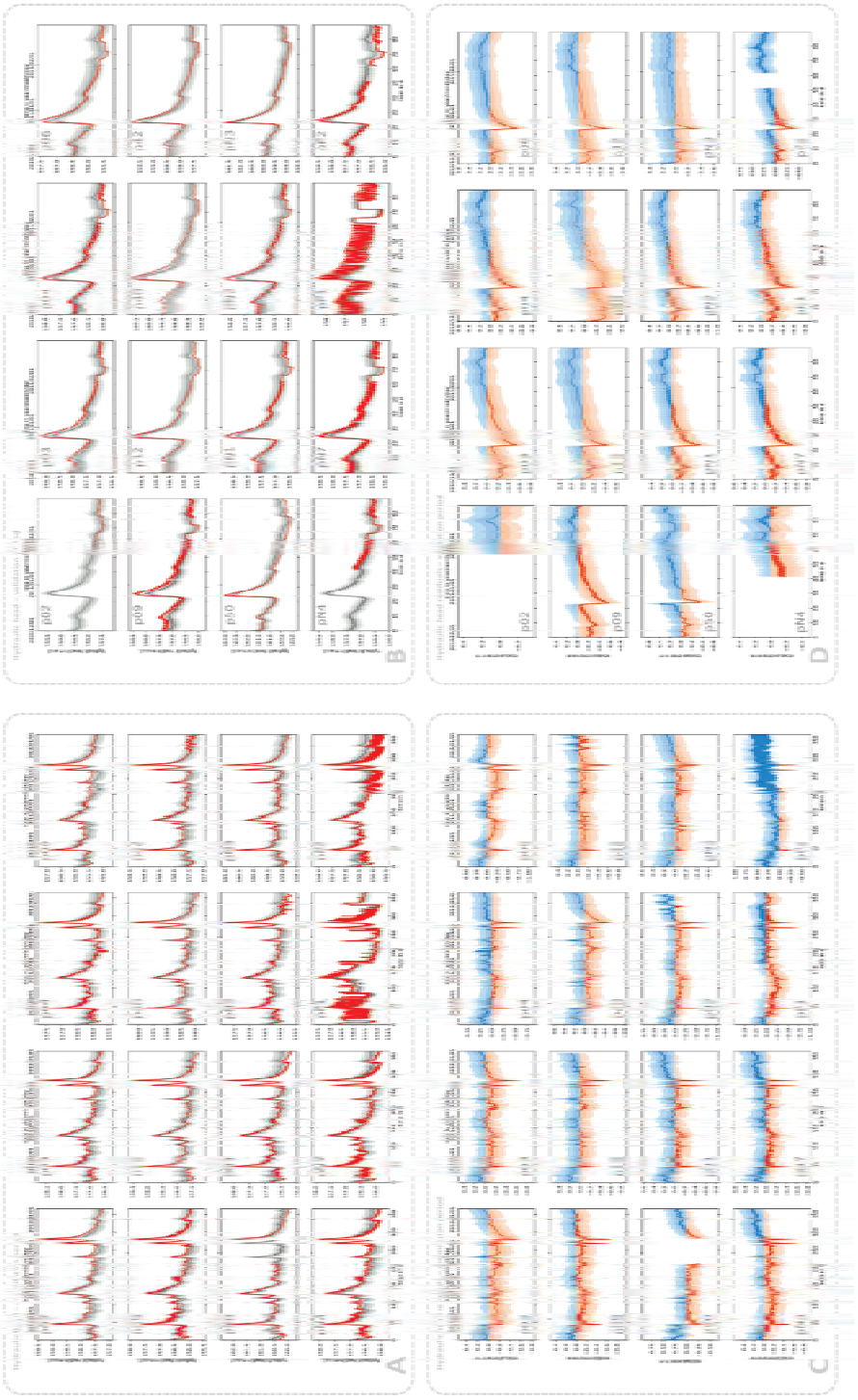


Figure II-11. Evaluation of the model predictions of Scenario 3c over the assimilation period (A, C) and validation period (B, D) for all observation and pumping wells. (A) and (B) illustrate the simulated mean (dark grey line) with single (grey area) and double (light grey area) lumped error standard deviation for the assimilation (A) and validation (B) periods. (C) and (D) illustrate the corresponding residual errors, colored according to positive (blue) and negative (red) residual components.

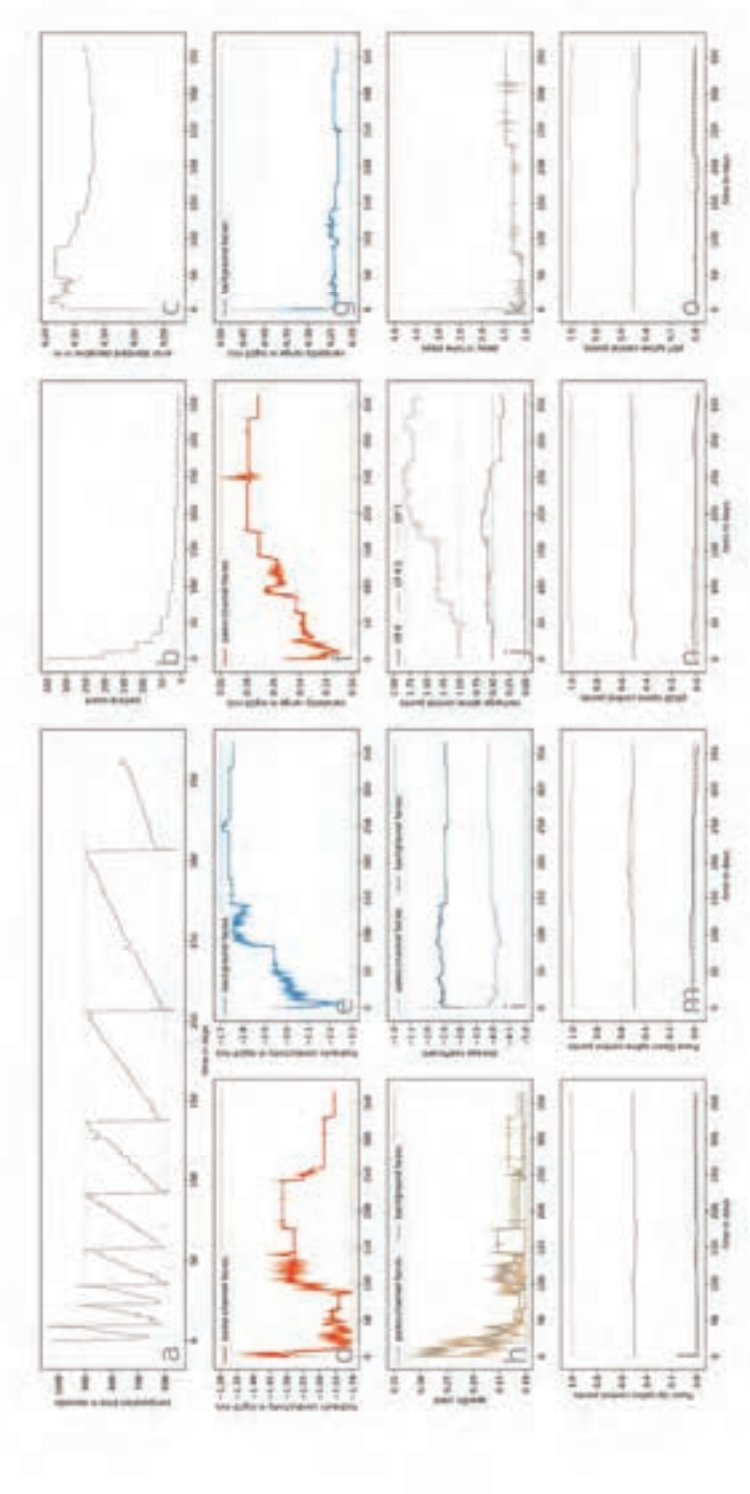


Figure II-12. Meta- and hyperparameters Scenario 1a over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

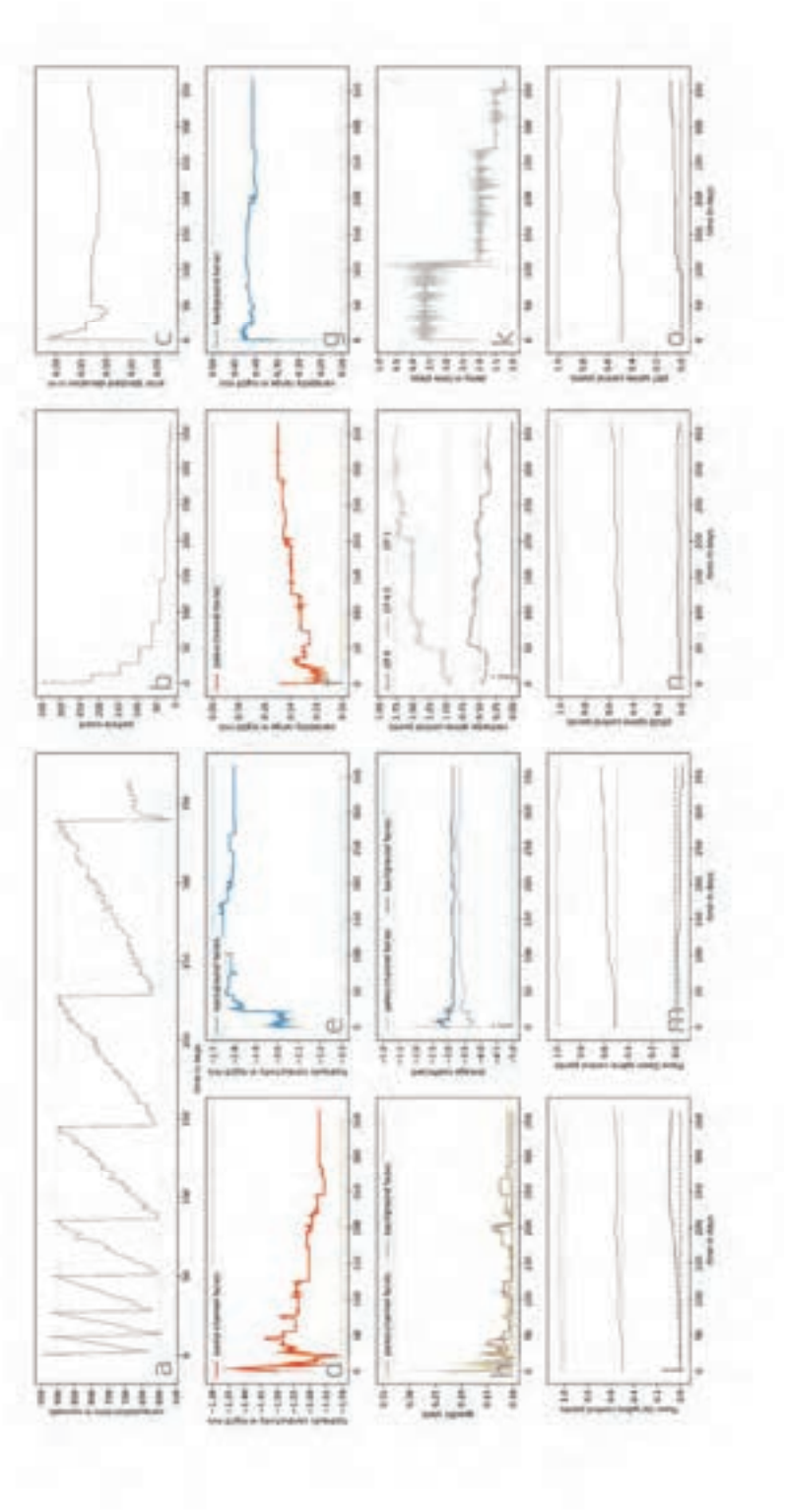


Figure II-13. Meta- and hyperparameters Scenario 1b over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

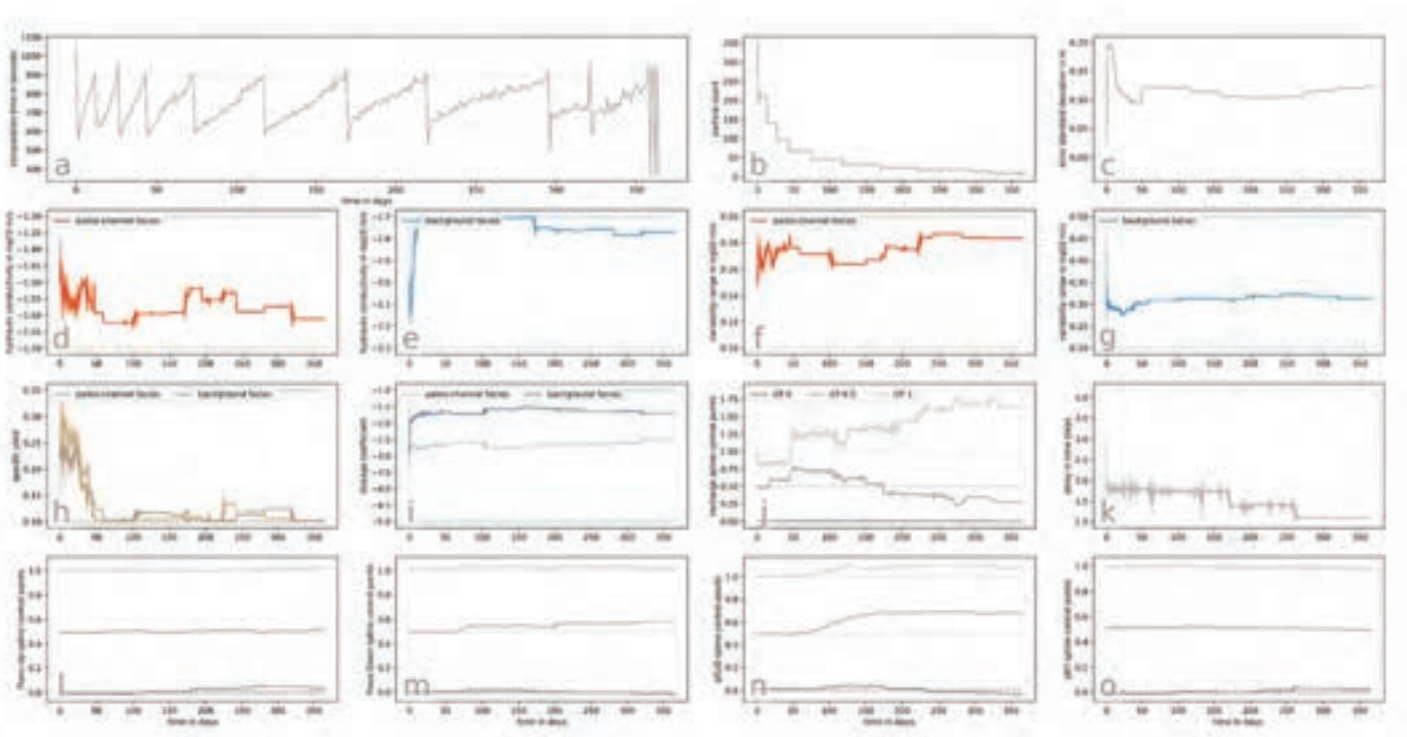


Figure II-14. Meta- and hyperparameters Scenario 1c over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

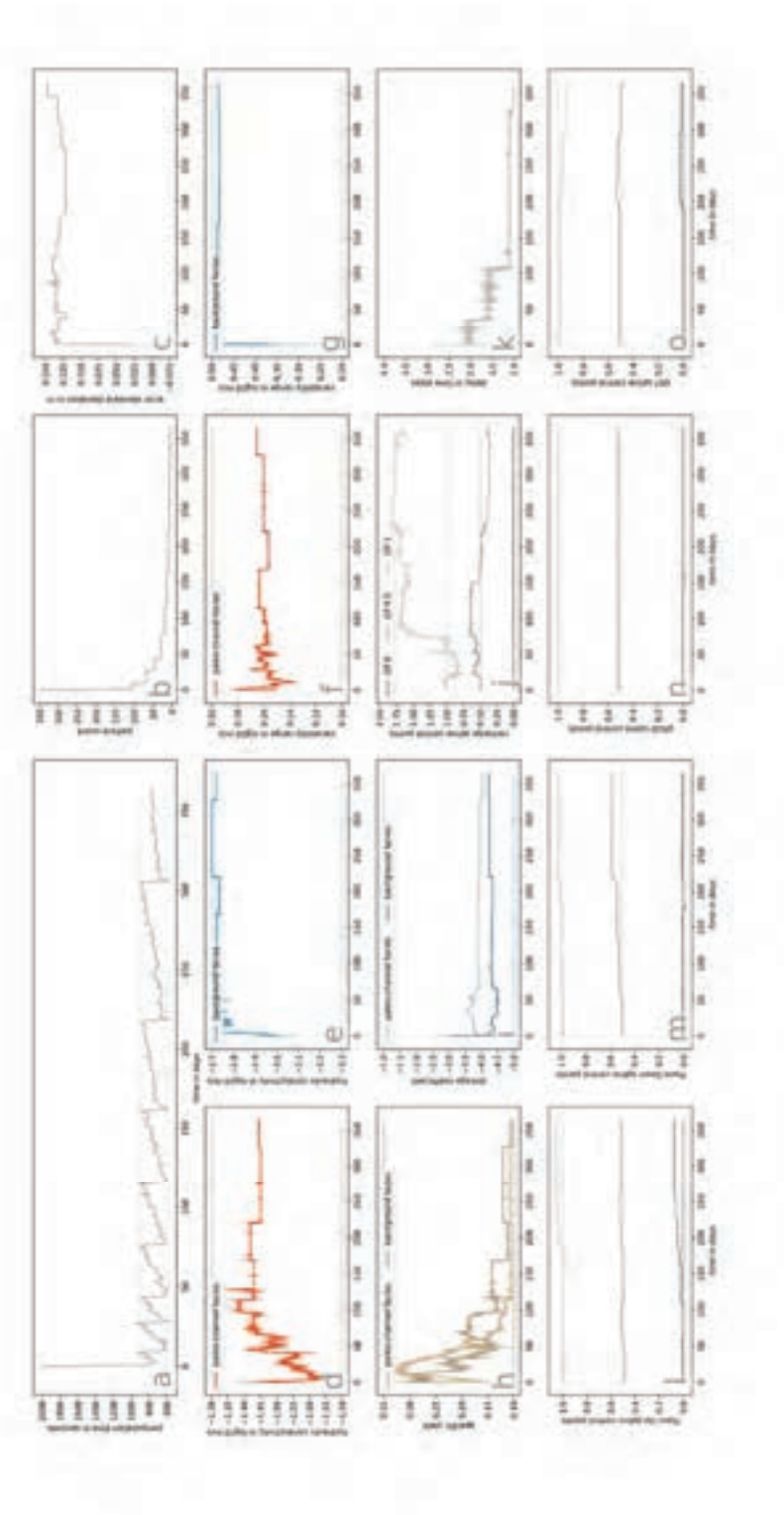


Figure II-15. Meta- and hyperparameters Scenario 2a over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

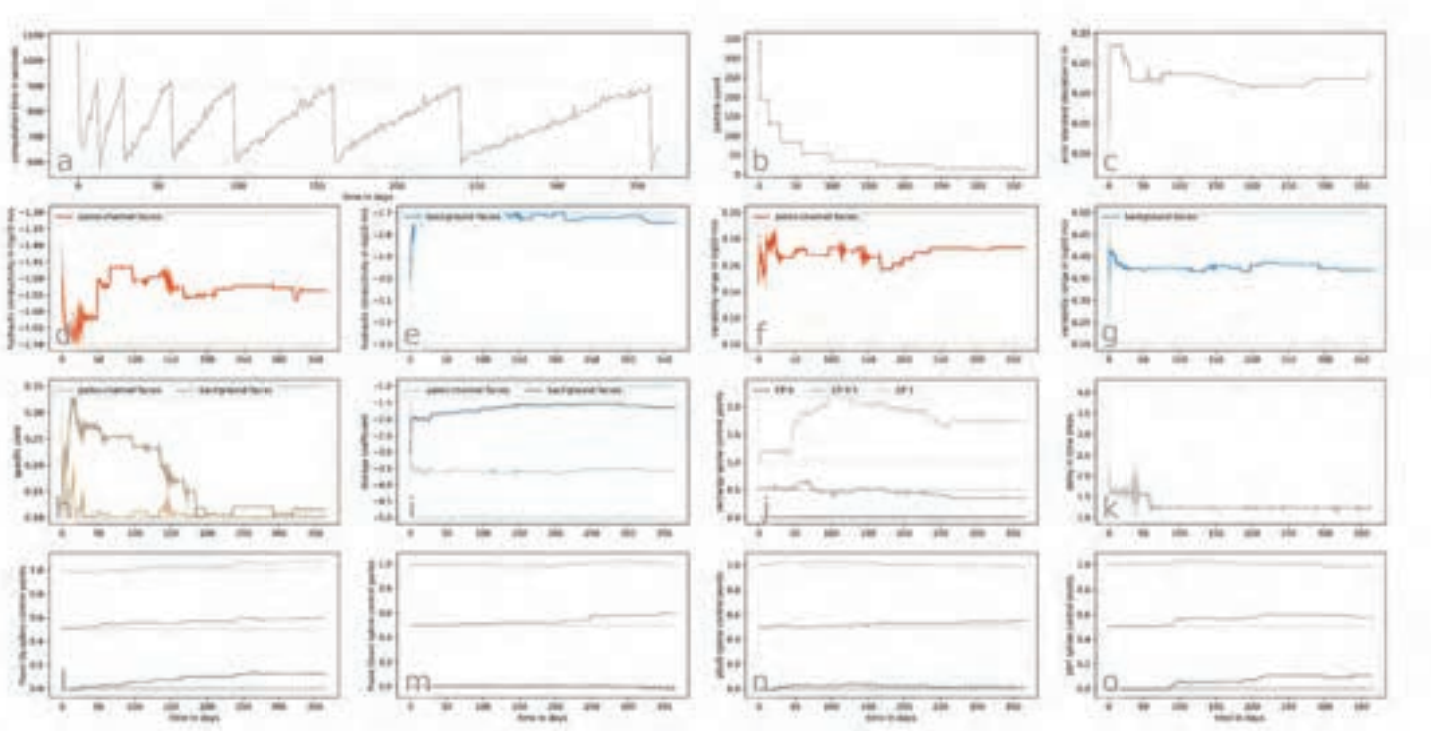


Figure II-16. Meta- and hyperparameters Scenario 2b over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

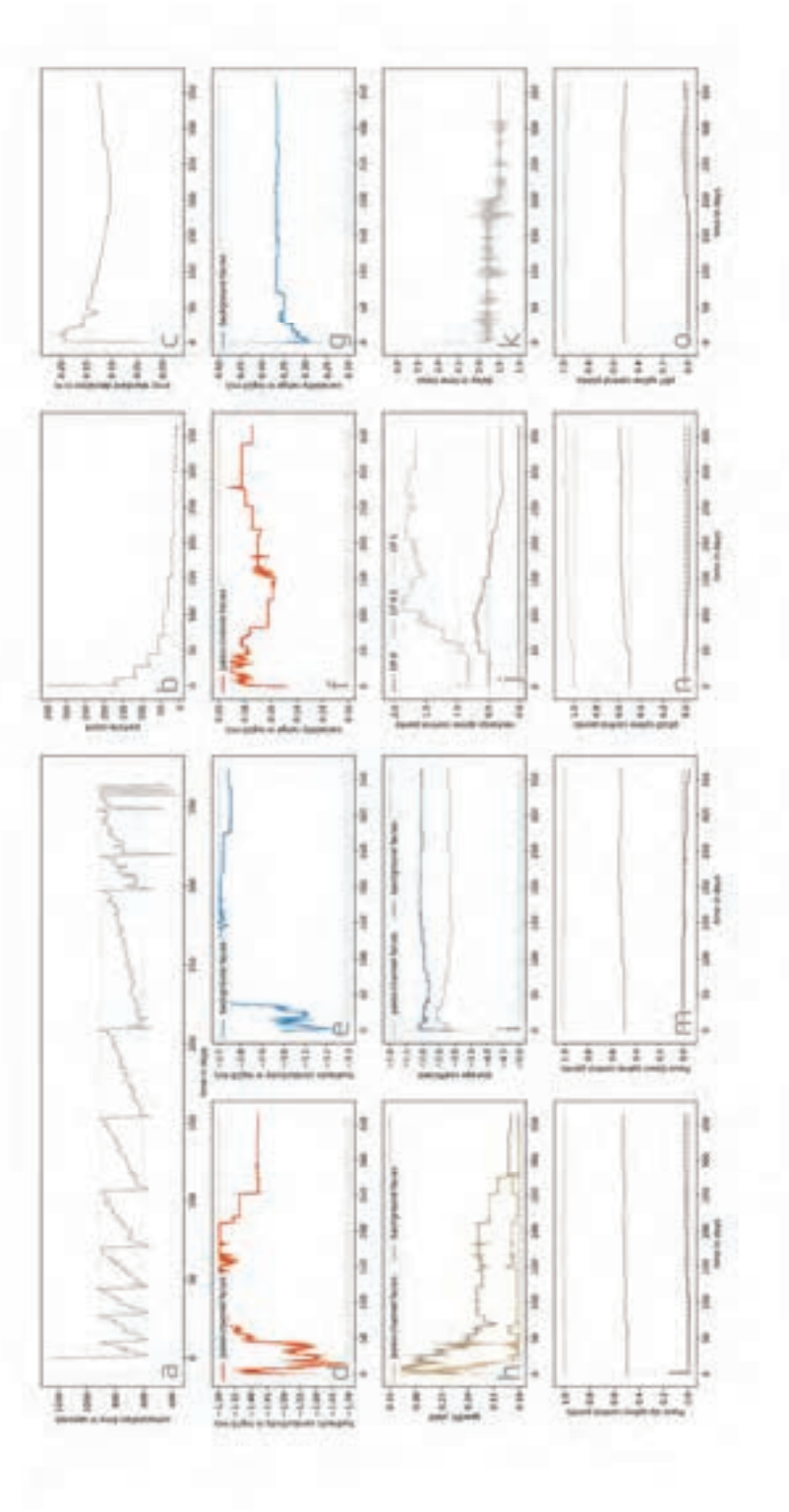


Figure II-17. Meta- and hyperparameters Scenario 2c over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

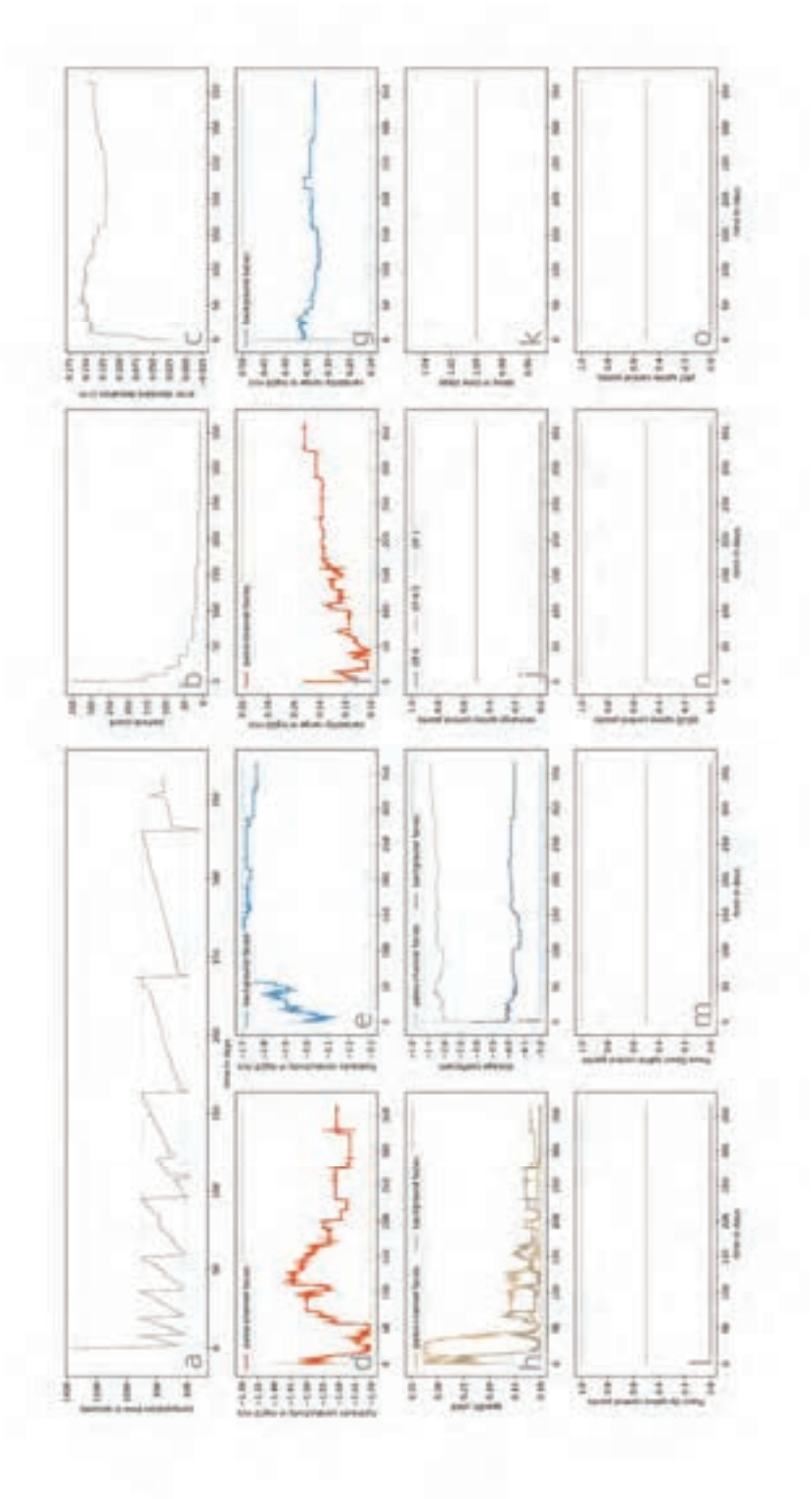


Figure II-18. Meta- and hyperparameters Scenario 3a over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

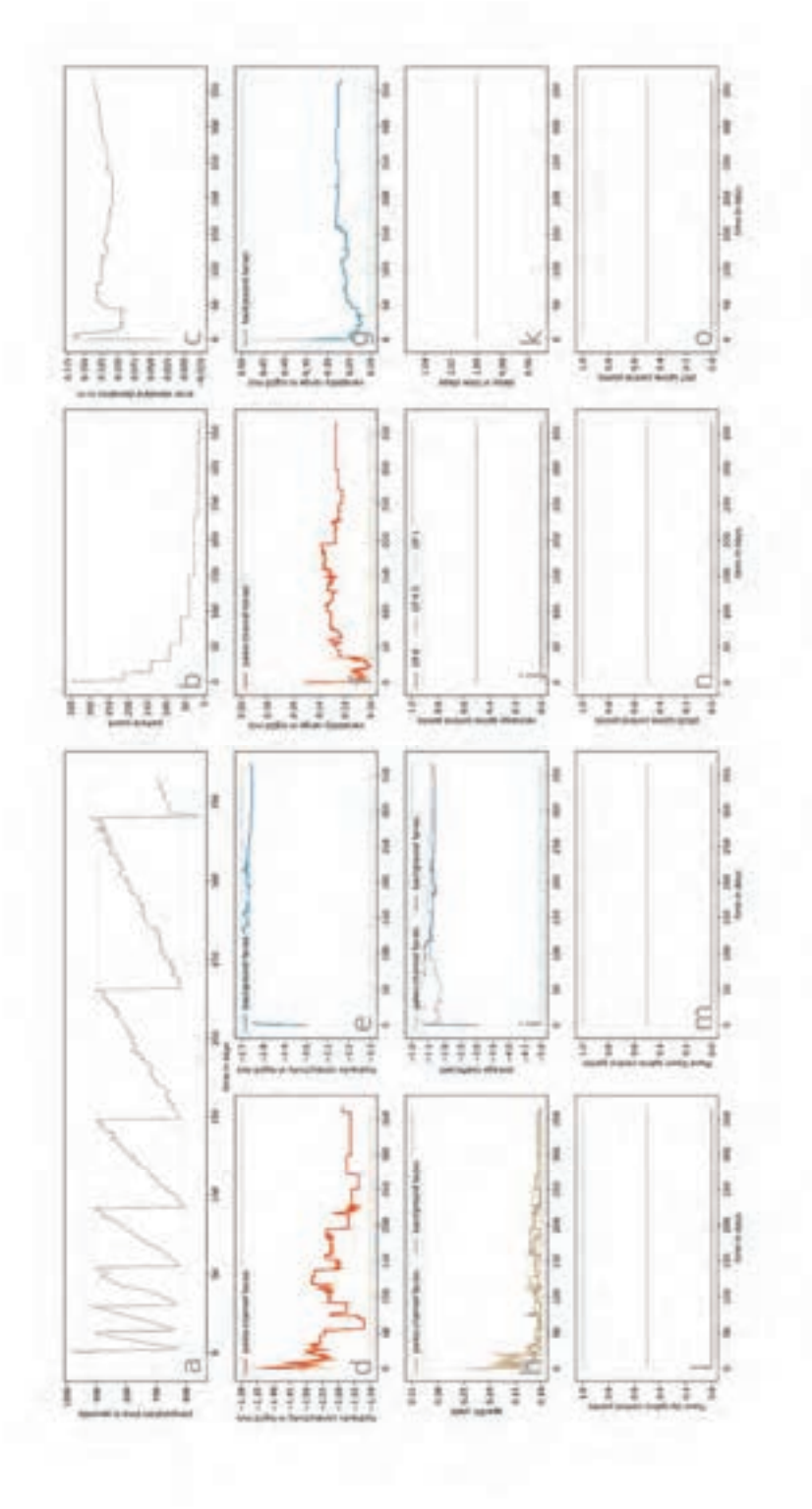


Figure II-19. Meta- and hyperparameters Scenario 3b over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

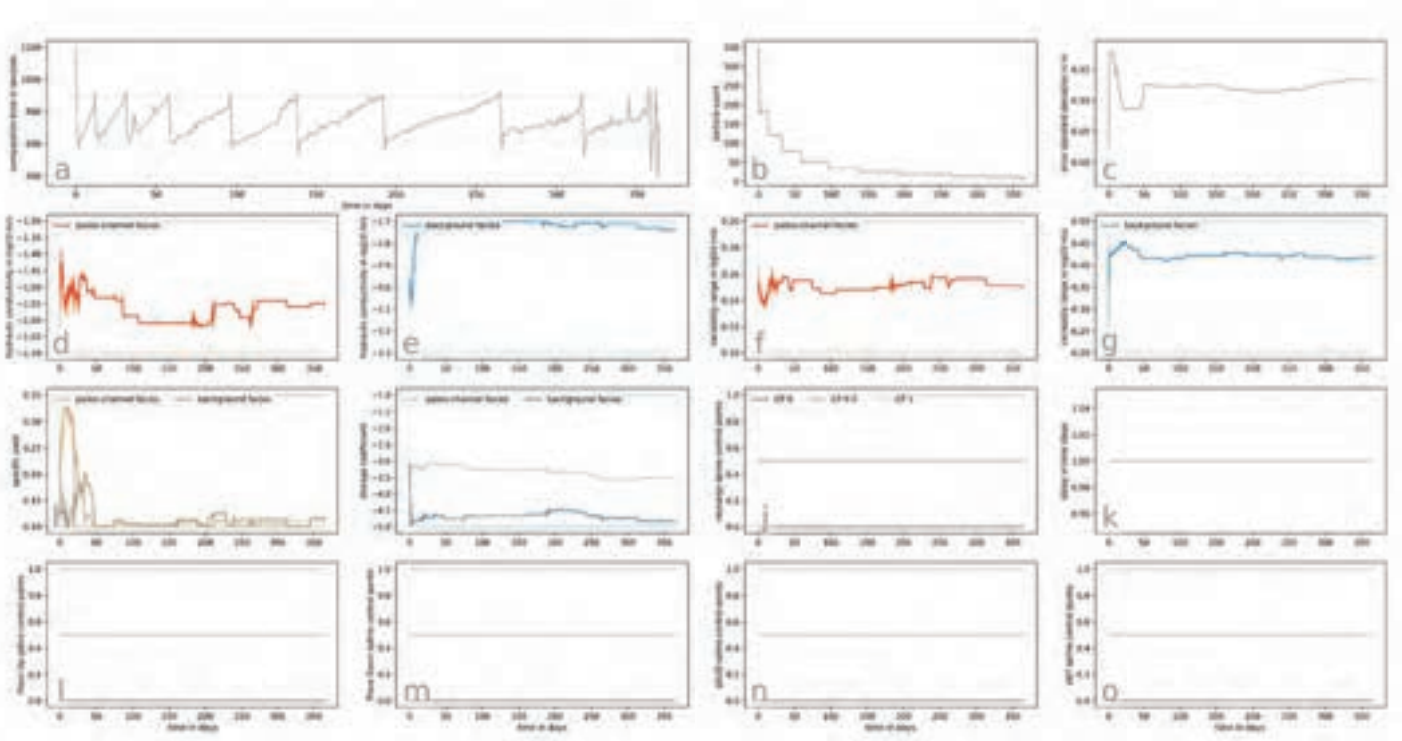


Figure II-20. Meta- and hyperparameters Scenario 3c over time: computation time per day (a), particle count (b), error standard deviation (c), mean hydraulic conductivity for facies 1 (d) and 2 (e), internal variability range for facies 1 (f) and 2 (g), specific yield for both facies (h), storage coefficient for both facies (i), recharge delay (j), and spline control points for recharge (k) and the boundary wells Piave Up (l), Piave Down (m), pSUD (n) and p07 (o). Full colors correspond to Scenario 1, darkened colors to Scenario 2, and lightened colors to Scenario 3. Single and double standard deviations of hyperparameter uncertainty are plotted in lighter shades, where applicable.

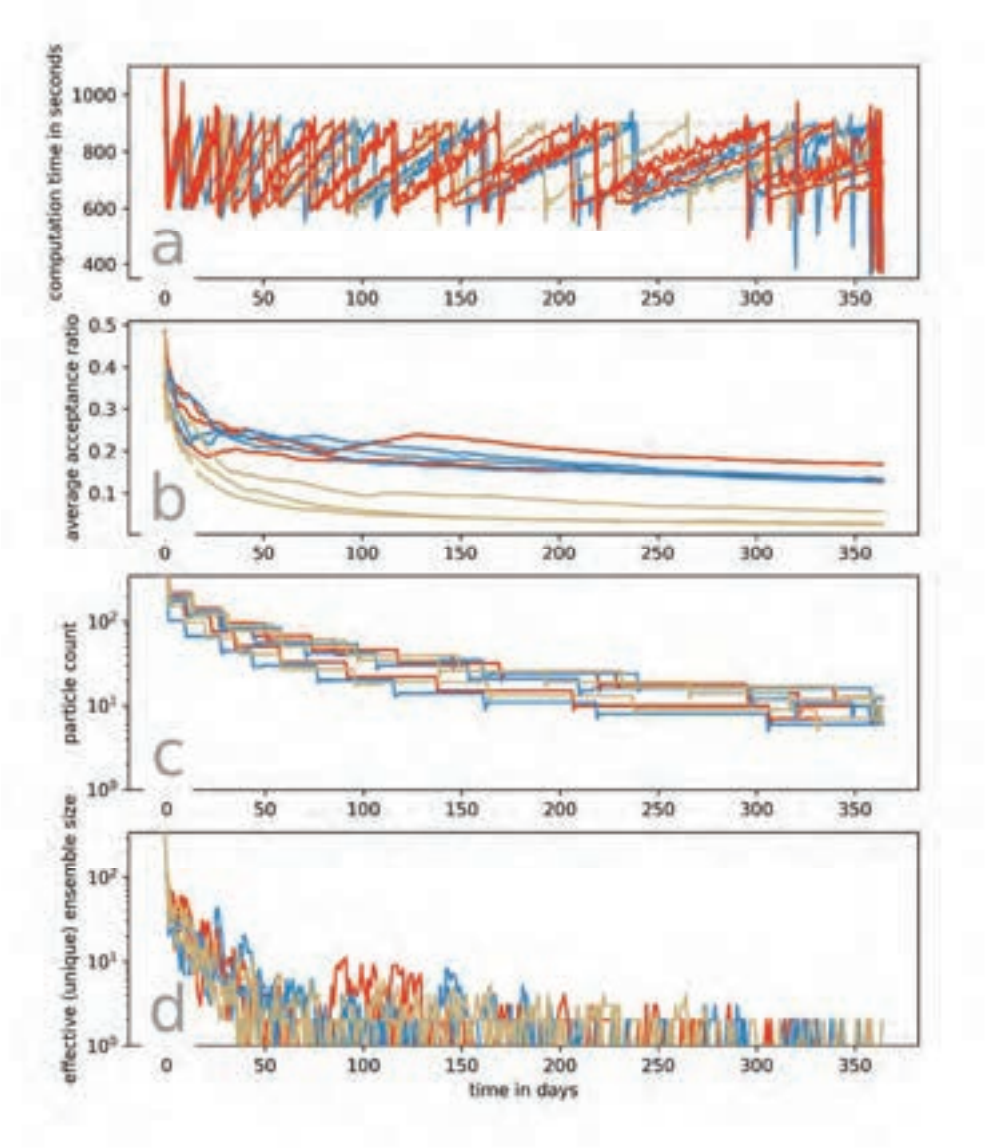


Figure II-21. Close-up of the simulation time (a), time-averaged MCMC proposal acceptance ratio (b), the raw particle count (c), and the effective, unique ensemble size (d) for Scenario 1 (red), Scenario 2 (blue), and Scenario 3 (brown).

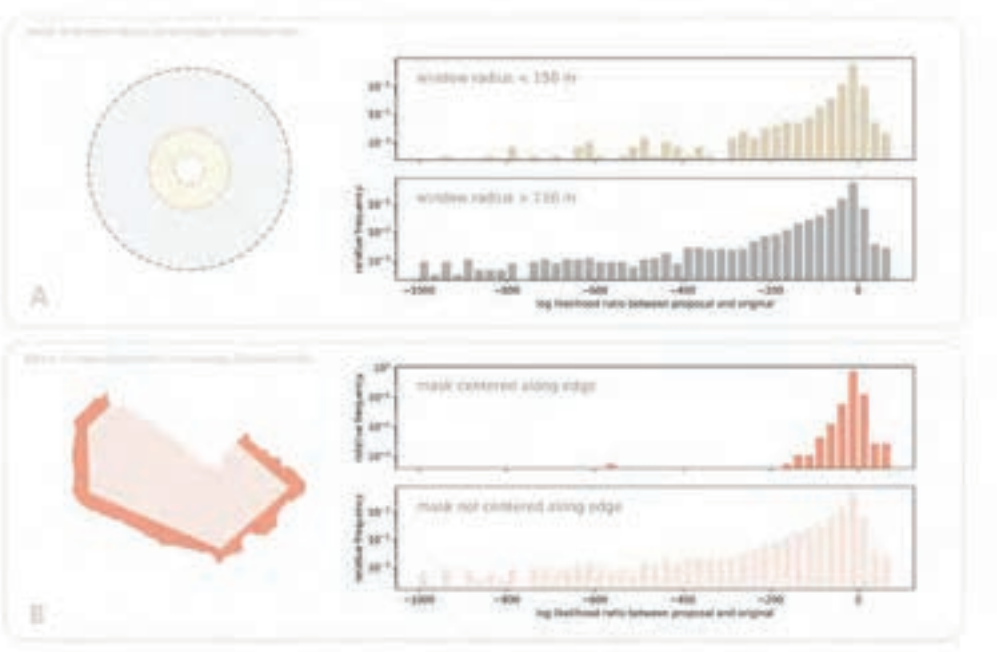


Figure II-22. Effect of window radius (A) and mask placement (B) on the average log-likelihood ratio (averaged down to likelihoods of a single time-step) between proposal and original, shown for Scenario 1c. Lower mask radii only have a minor effect on the magnitude of this ratio (A), whereas placing the mask near the boundaries significantly reduces the magnitude of the likelihood ratios.

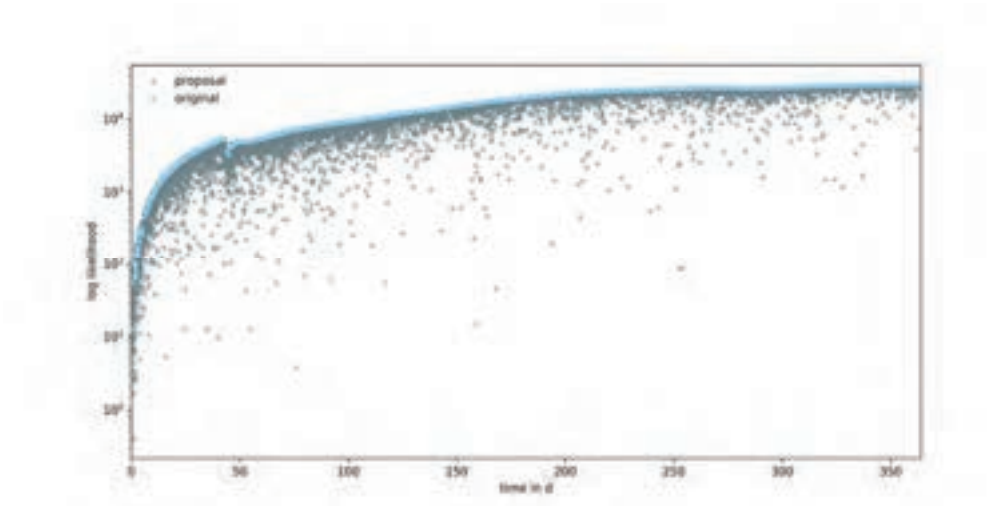


Figure II-23. Total likelihoods of the originals (light blue) and the proposals (grey), for Scenario 1c. The majority of proposals do not constitute improvements. Proposals with negative total likelihoods are omitted from this plot.

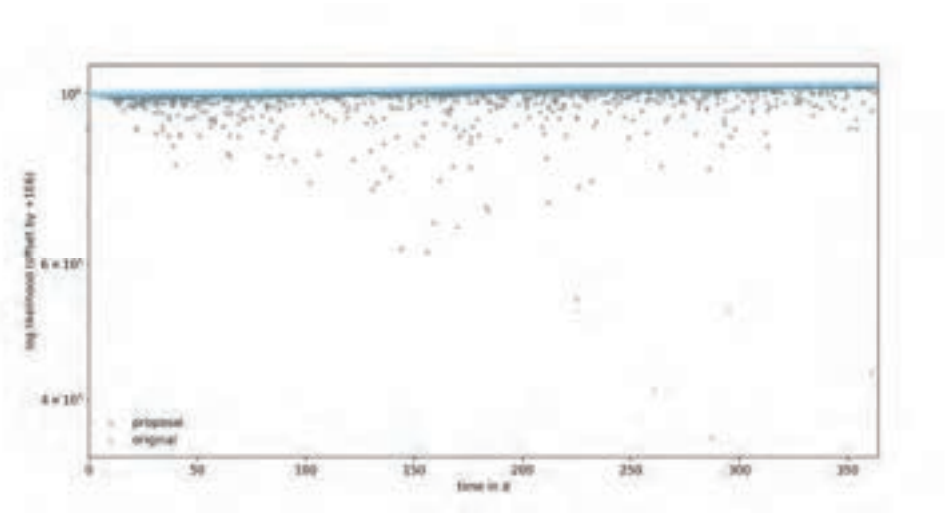


Figure II-24. Total likelihoods of the originals (light blue) and the proposals (grey), for Scenario 1c. The total likelihoods are shifted by 1.0E6 to also depict outliers omitted in Figure II-23.

Table II-1. List of symbols in the manuscript.

Symbol		Type	Meaning
$\boldsymbol{x}$		*vector*	state predictions
$\boldsymbol{\theta}$		*vector*	parameters (Section 1, Section 2); grid parameters (Section 3 onwards)
$\boldsymbol{y}$		*vector*	state observations (observed or simulated)
$\boldsymbol{\Theta}$		*vector*	hyperparameters
$\boldsymbol{U}$		*vector*	raw forcing data
$\boldsymbol{u}$		*vector*	transformed forcings
$\boldsymbol{p}(\cdot)$		*function*	probability density function (pdf)
$\widehat{\boldsymbol{p}}(\cdot)$		*function*	particle approximation of probability density function
$\boldsymbol{\delta}(\cdot)$		*function*	Dirac delta measure
$\boldsymbol{H}$		*matrix*	observation extraction matrix
$\boldsymbol{\mu}$		*vector*	mean
$\boldsymbol{\Sigma}$		*matrix*	covariance matrix
$\boldsymbol{N}$		*scalar*	number of particles
$\boldsymbol{n}$		*scalar*	particle index
$\boldsymbol{a}$		*scalar*	resampled particle index
$\boldsymbol{w}$		*scalar*	unnormalized particle weight
$\boldsymbol{W}$		*scalar*	normalized particle weight
$\boldsymbol{l}$		*scalar*	likelihood increment
$\boldsymbol{L}$		*scalar*	total likelihood
$\boldsymbol{N_{cell}}$		*scalar*	number of model cells
$\boldsymbol{\mathcal{U}}(\cdot)$		*function*	uniform probability density function
$\boldsymbol{\mathcal{N}}(\cdot)$		*function*	Gaussian probability density function
$\boldsymbol{SUR}(\cdot)$		*function*	stochastic universal resampling
$\boldsymbol{G}(\cdot)$		*function*	field generator
$\boldsymbol{F}(\cdot)$		*function*	forcing model
$\boldsymbol{IDW}(\cdot)$		*function*	inverse distance weighting
$\boldsymbol{M}(\cdot)$		*function*	numerical model
$\boldsymbol{K}$		*vector*	hydraulic conductivity
$\boldsymbol{\Delta K}$		*vector*	internal variability of hydraulic conductivity
$\boldsymbol{S_y}$		*vector*	specific yield
$\boldsymbol{SCP}$		*vector*	forcing spline control points
$\boldsymbol{\lambda}$		*scalar*	recharge delay
$\boldsymbol{\sigma}$		*scalar*	lumped model error standard deviation

Table II-2. Performance metrics for the posterior ensemble of hyperparameters at the end of the 1-year assimilation period, evaluated over the assimilation period, the 90-day validation period, and the shortened 60-day validation period. The lower half of the table lists the 【RMSE】^os over the assimilation period for all observation wells. Continued on next page.

	assimilation period					validation period					validation period short				
Scenario	$\overline{RMSE}$ [m]	$\overline{bias}$ [m]	$\overline{pbias}$ [%]	$\overline{KGE}$ [-]	$\overline{r_{sp}}$ [-]	$\overline{RMSE}$ [m]	$\overline{bias}$ [m]	$\overline{pbias}$ [%]	$\overline{KGE}$ [-]	$\overline{r_{sp}}$ [-]	$\overline{RMSE}$ [m]	$\overline{bias}$ [m]	$\overline{pbias}$ [%]	$\overline{KGE}$ [-]	$\overline{r_{sp}}$ [-]
1a	0.134	0.000	0.003	0.910	0.921	0.142	-0.003	0.002	0.828	0.982	0.145	-0.067	-0.175	0.866	0.986
1b	0.134	-0.003	-0.007	0.913	0.923	0.138	0.003	0.016	0.848	0.985	0.139	-0.054	-0.139	0.879	0.988
1c	0.125	-0.002	-0.006	0.925	0.935	0.128	-0.005	-0.002	0.860	0.985	0.131	-0.056	-0.142	0.884	0.983
2a	0.144	0.005	0.016	0.897	0.908	0.168	0.074	0.216	0.810	0.982	0.138	0.007	0.030	0.836	0.982
2b	0.129	-0.004	-0.010	0.926	0.917	0.145	0.041	0.125	0.853	0.980	0.134	-0.017	-0.034	0.886	0.986
2c	0.128	0.001	0.005	0.909	0.928	0.139	-0.008	-0.010	0.854	0.984	0.147	-0.063	-0.162	0.881	0.983
3a	0.143	0.001	0.009	0.873	0.909	0.158	0.008	0.035	0.768	0.980	0.157	-0.063	-0.160	0.774	0.988
3b	0.138	0.006	0.020	0.857	0.920	0.165	-0.024	-0.054	0.766	0.977	0.178	-0.102	-0.266	0.792	0.979
3c	0.138	-0.001	0.000	0.862	0.910	0.153	-0.006	-0.007	0.767	0.980	0.156	-0.079	-0.206	0.778	0.984

continuation

Scenario	$RMSE^o$ [m] over the assimilation period															
	p02	p03	p04	p06	p09	p12	p40	p42	p50	pN1	pN2	pN3	pN4	pN7	pP1	pP2
1a	0.107	0.129	0.085	0.102	0.119	0.091	0.108	0.086	0.121	0.120	0.115	0.074	0.131	0.120	0.134	0.318
1b	0.095	0.104	0.085	0.154	0.101	0.104	0.093	0.097	0.117	0.102	0.136	0.079	0.132	0.120	0.125	0.315
1c	0.101	0.089	0.069	0.086	0.113	0.114	0.105	0.120	0.121	0.098	0.096	0.073	0.119	0.093	0.117	0.304
2a	0.126	0.138	0.109	0.112	0.119	0.093	0.150	0.103	0.129	0.115	0.112	0.098	0.130	0.132	0.122	0.339
2b	0.103	0.091	0.098	0.108	0.115	0.096	0.123	0.097	0.123	0.120	0.117	0.080	0.157	0.122	0.122	0.276
2c	0.088	0.104	0.077	0.121	0.106	0.082	0.101	0.096	0.131	0.097	0.102	0.083	0.122	0.113	0.114	0.317
3a	0.093	0.095	0.114	0.120	0.144	0.111	0.106	0.112	0.131	0.136	0.143	0.081	0.137	0.128	0.139	0.327
3b	0.099	0.110	0.095	0.109	0.109	0.095	0.114	0.096	0.130	0.143	0.120	0.084	0.132	0.120	0.133	0.330
3c	0.135	0.105	0.101	0.154	0.107	0.092	0.109	0.100	0.129	0.122	0.117	0.076	0.140	0.125	0.133	0.312

Table II-3. Mean α for the calculation of the $\overline{KGE}$ for all scenarios and simulation periods.

α	assimilation period	validation period	validation period short
Scenario 1	0.94	0.85	0.88
Scenario 2	0.94	0.84	0.87
Scenario 3	0.88	0.77	0.78

Table II-4. Mean β for the calculation of the $\overline{KGE}$ for all scenarios and simulation periods.

β	assimilation period	validation period	validation period short
Scenario 1	1.00	1.01	1.01
Scenario 2	1.00	1.01	1.01
Scenario 3	1.00	1.01	1.00

Table II-5. Mean r for the calculation of the $\overline{KGE}$ for all scenarios and simulation periods.

r	assimilation period	validation period	validation period short
Scenario 1	0.96	0.98	0.98
Scenario 2	0.96	0.98	0.97
Scenario 3	0.95	0.98	0.98

III Supporting Information for Chapter 4

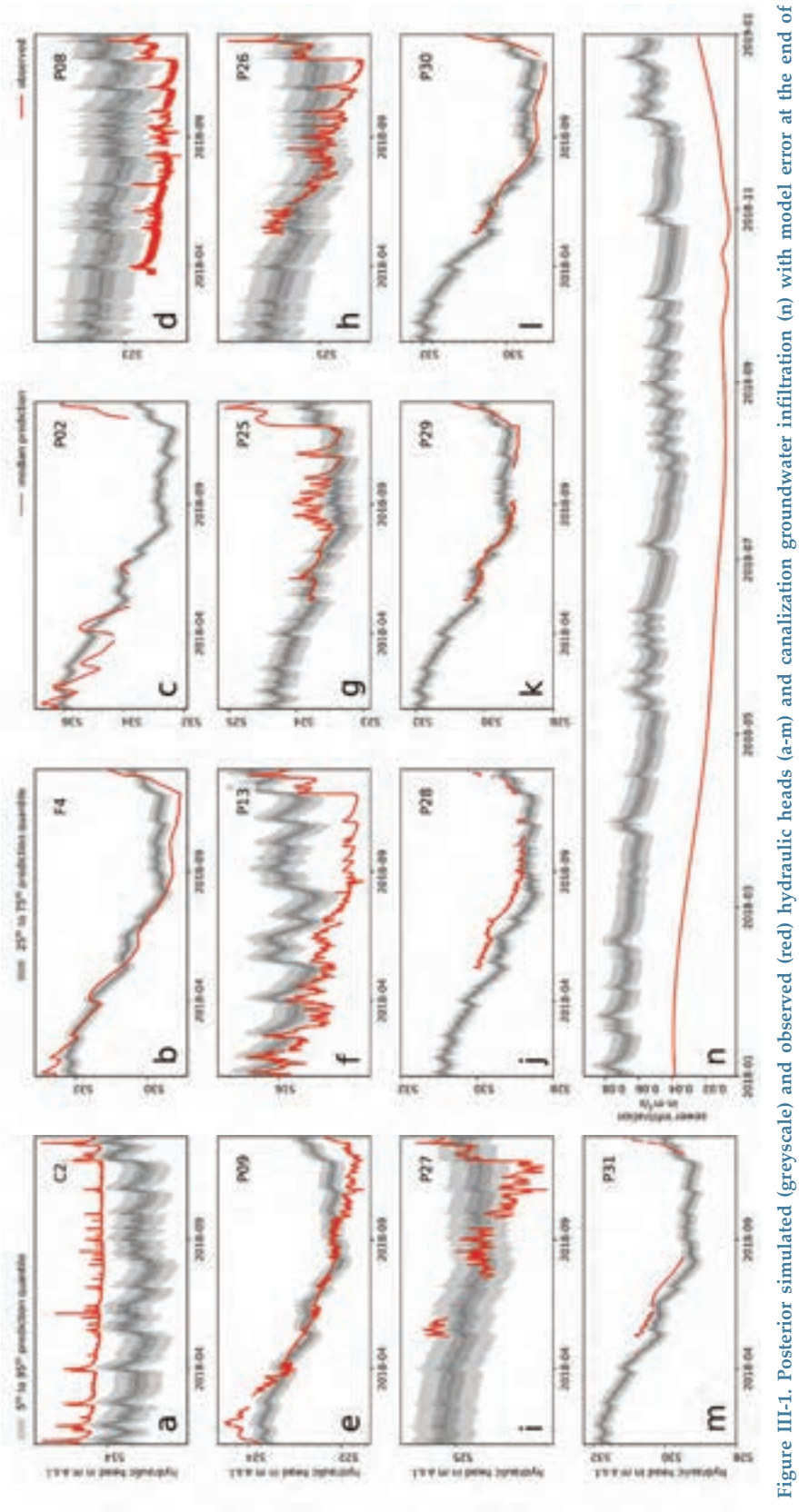


Figure III-1. Posterior simulated (greyscale) and observed (red) hydraulic heads (a-m) and canalization groundwater infiltration (n) with model error at the end of simulation period for $N = 30$.

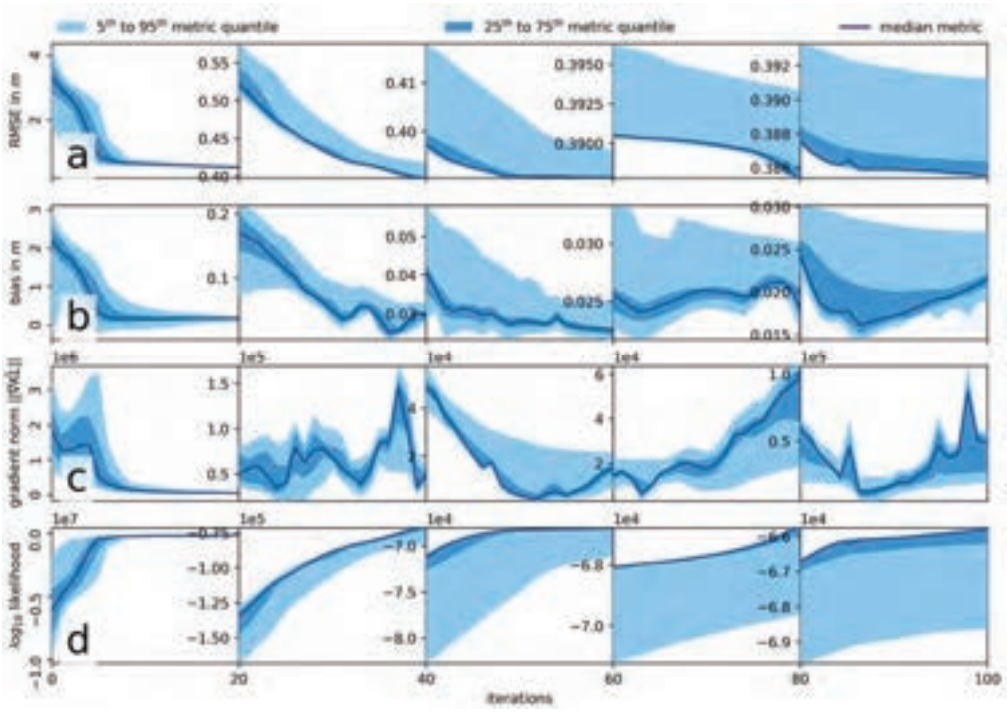


Figure III-2. Posterior overall root-mean square error (o) and bias (p) for the hydraulic heads, the mean norm of the Kullback-Leibler divergence gradient (q) and the log-likelihood (r) across the algorithm's iterations for the scenario $N = 30$. For better visualization, the y-axis scale is reset every 20 iterations.

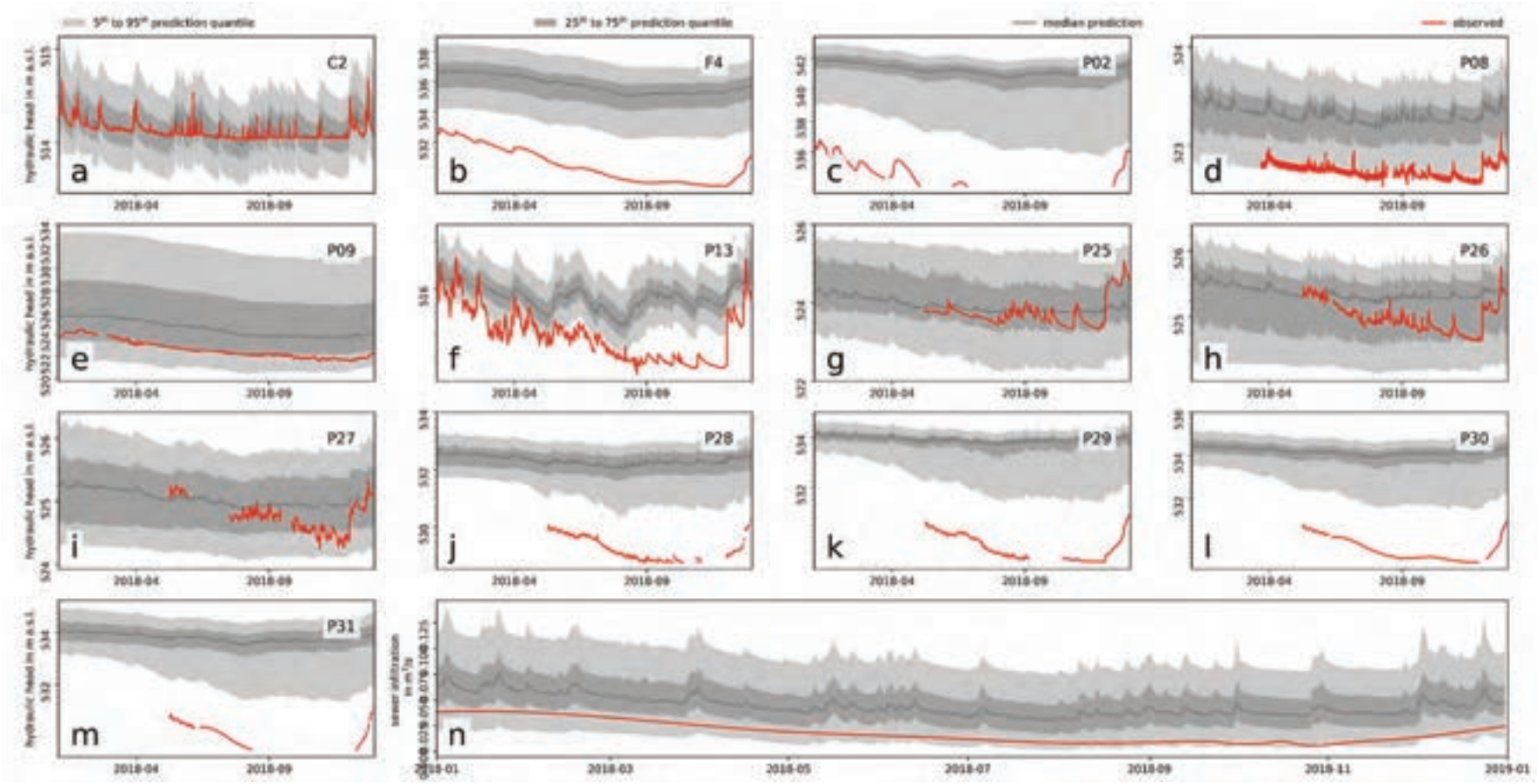


Figure III-3. Prior simulated (greyscale) and observed (red) hydraulic heads (a-m) and canalization groundwater infiltration (n) with model error at the end of simulation period for $N = 100$.

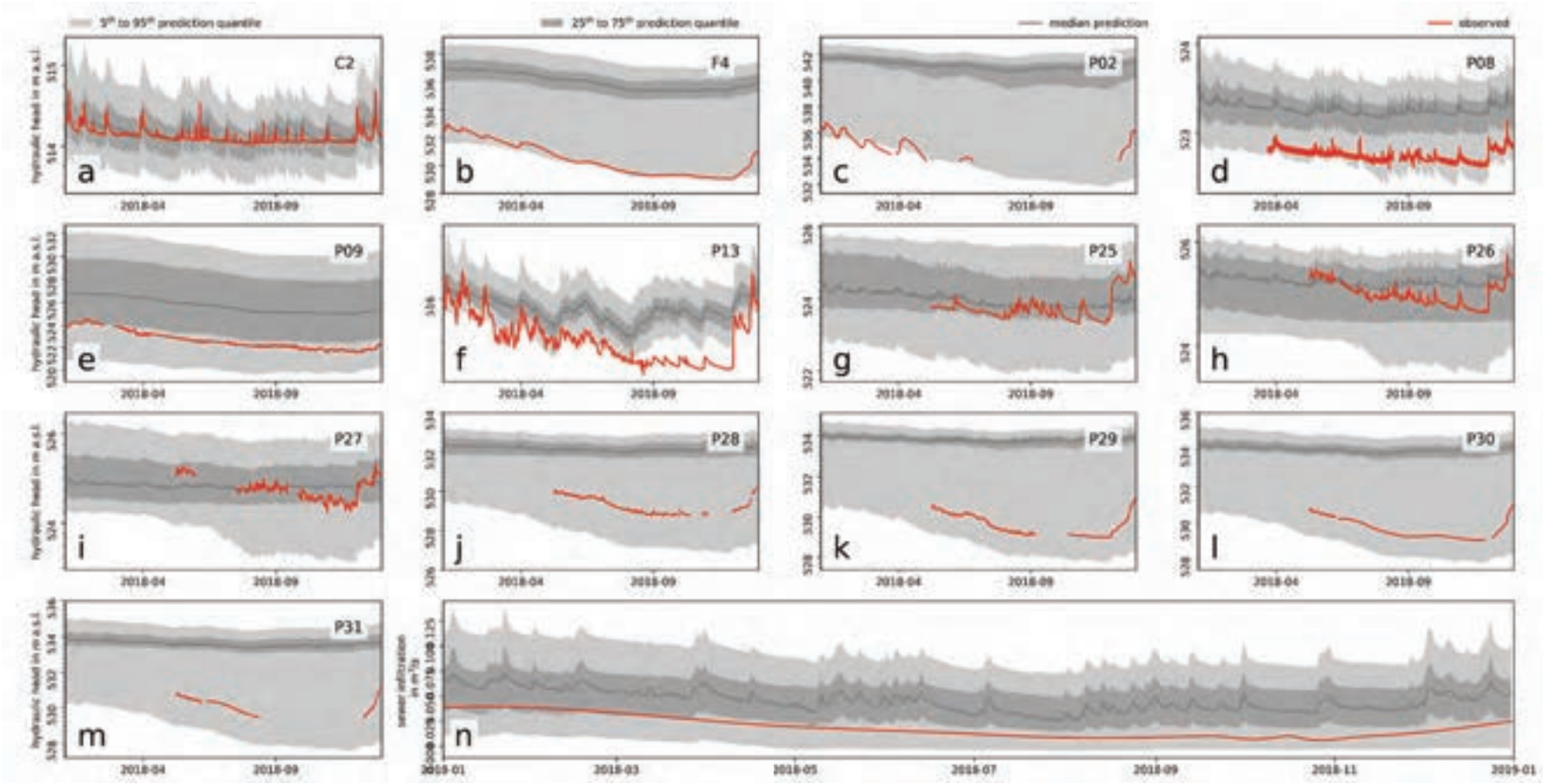


Figure III-4. Prior simulated (greyscale) and observed (red) hydraulic heads (a-m) and canalization groundwater infiltration (n) with model error at the end of simulation period for $N = 30$.

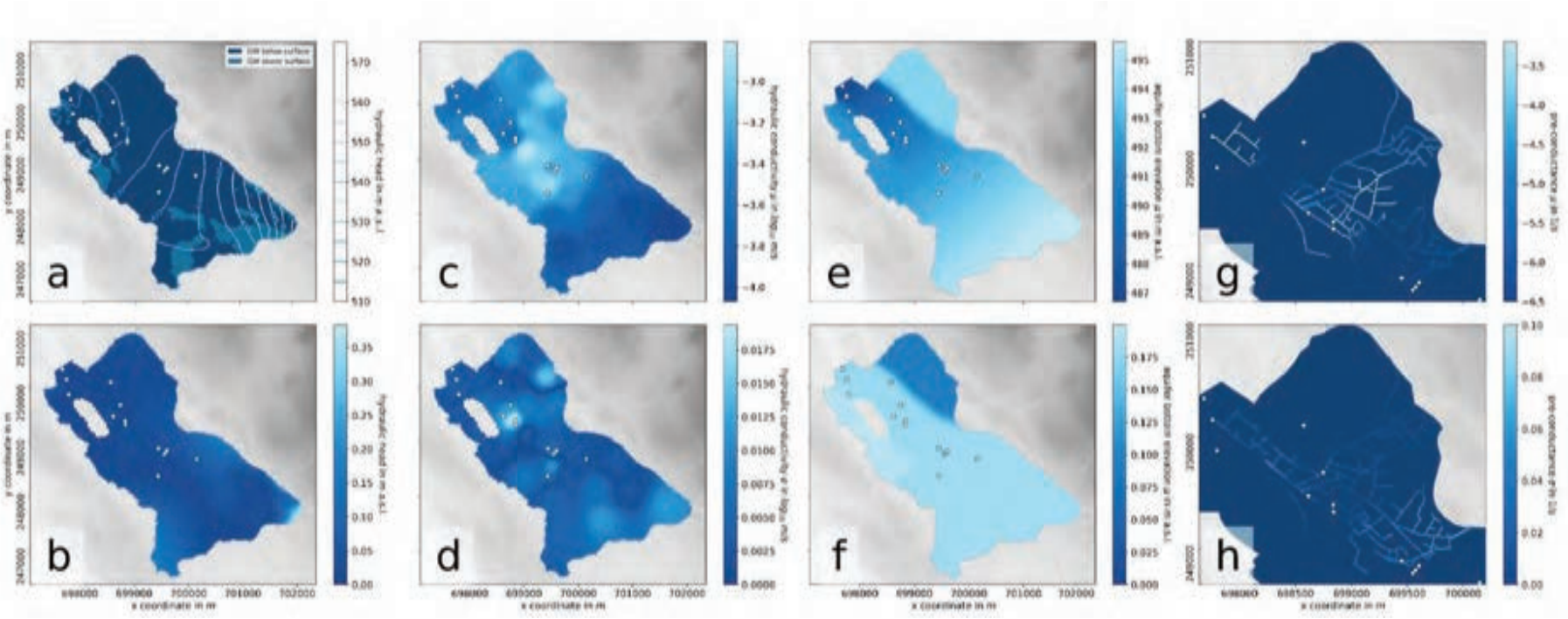


Figure III-5. Posterior parameters and hydraulic head at the final iteration for $N = 30$. The two rows illustrate mean (a, c, e, g) and standard deviations (b, d, f, h) of hydraulic head in the initial steady-state simulation period (c, d), hydraulic conductivity (e, f), aquifer bottom elevation (g, h), and canalization conductance (i, j).

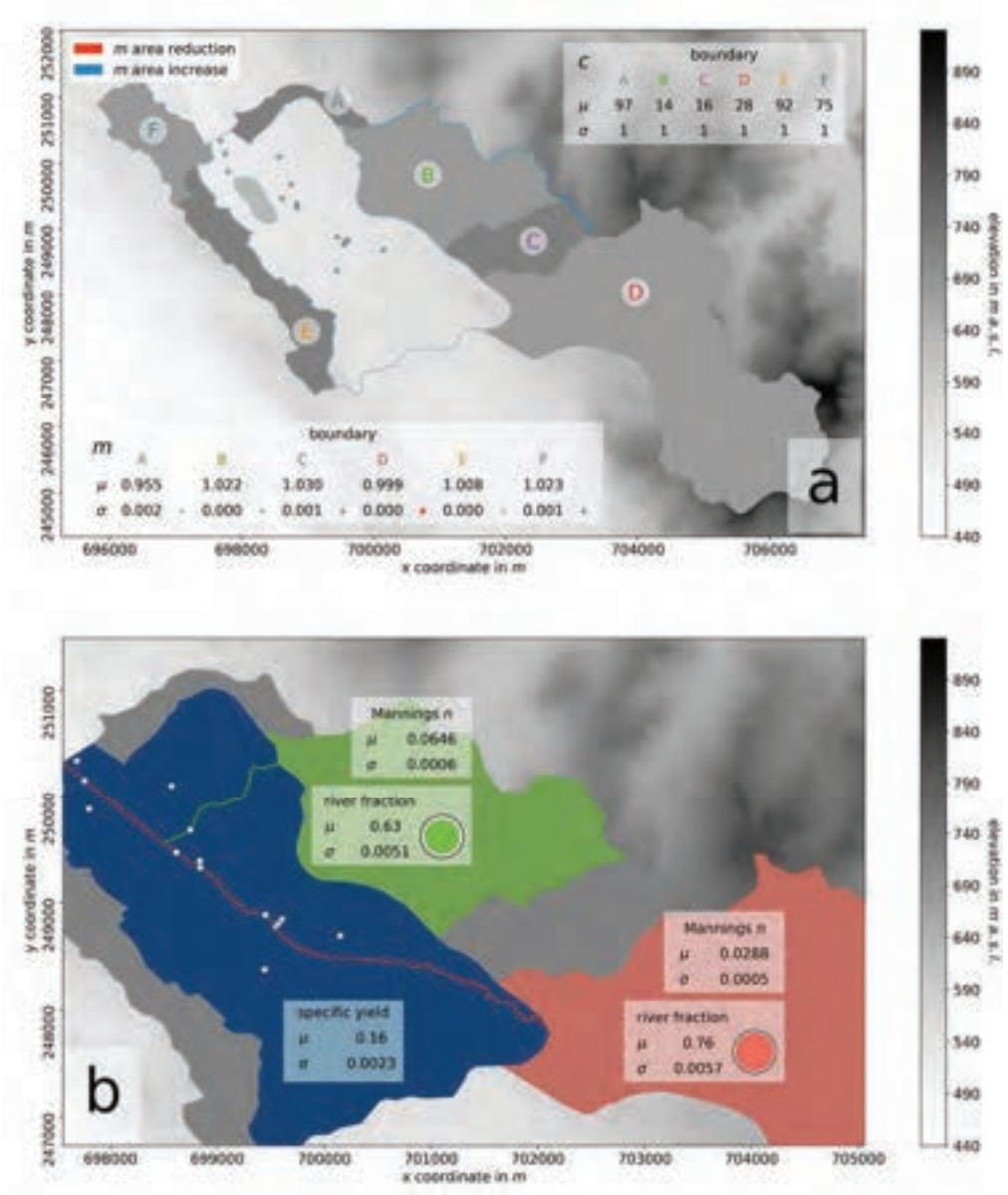


Figure III-6. Posterior values for specific yield and the forcing-related parameters at the final iteration for $N = 30$. Subplot (a) illustrates the boundary-related parameters: the scalar multiplier m and the recharge delay factor d in hours; mean m is illustrated as a faint outline around the boundaries, visualizing the area inflation or deflation. Subplot (b) shows the specific yield and river-related parameters: the river discharge fraction and the Manning's number for Wildbach and Luppmen, respectively.

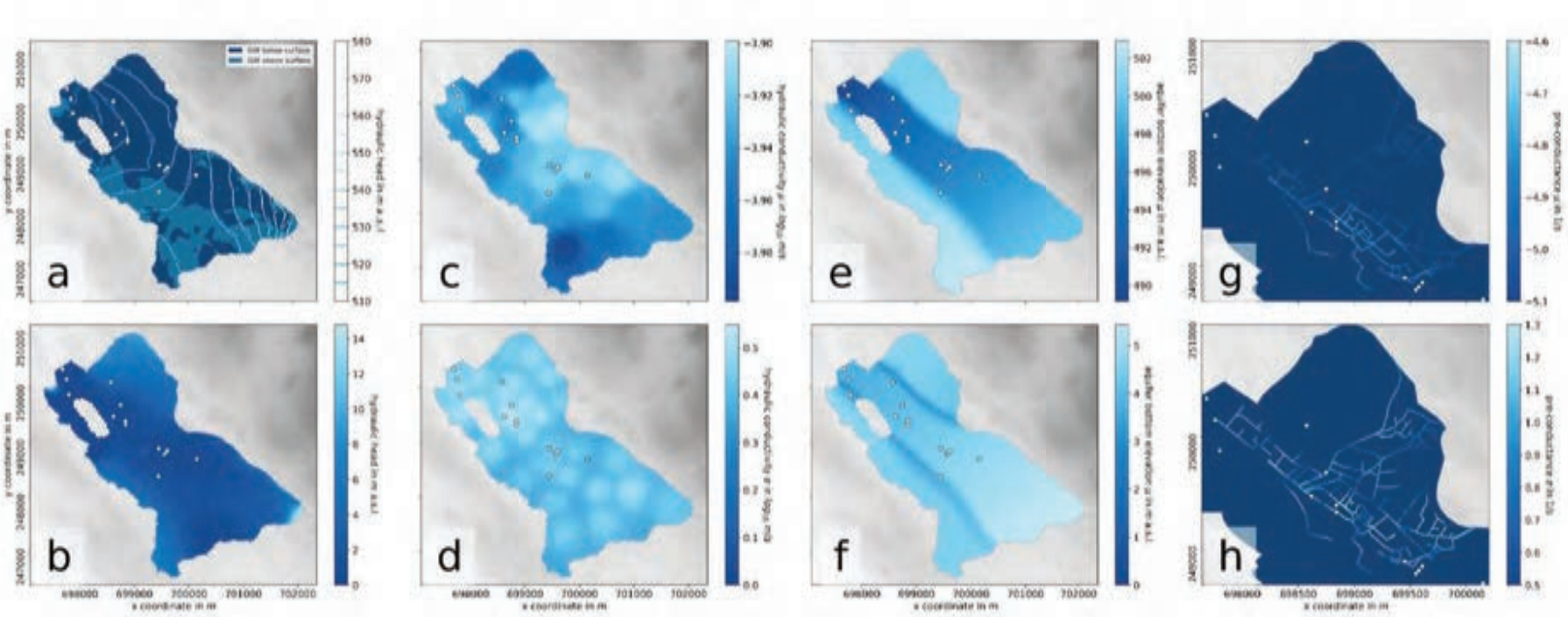


Figure III-7. Prior parameters and hydraulic head for $N = 100$. The two rows illustrate mean (a, c, e, g) and standard deviations (b, d, f, h) of hydraulic head in the initial steady-state simulation period (c, d), hydraulic conductivity (e, f), aquifer bottom elevation (g, h), and canalization conductance (i, j).

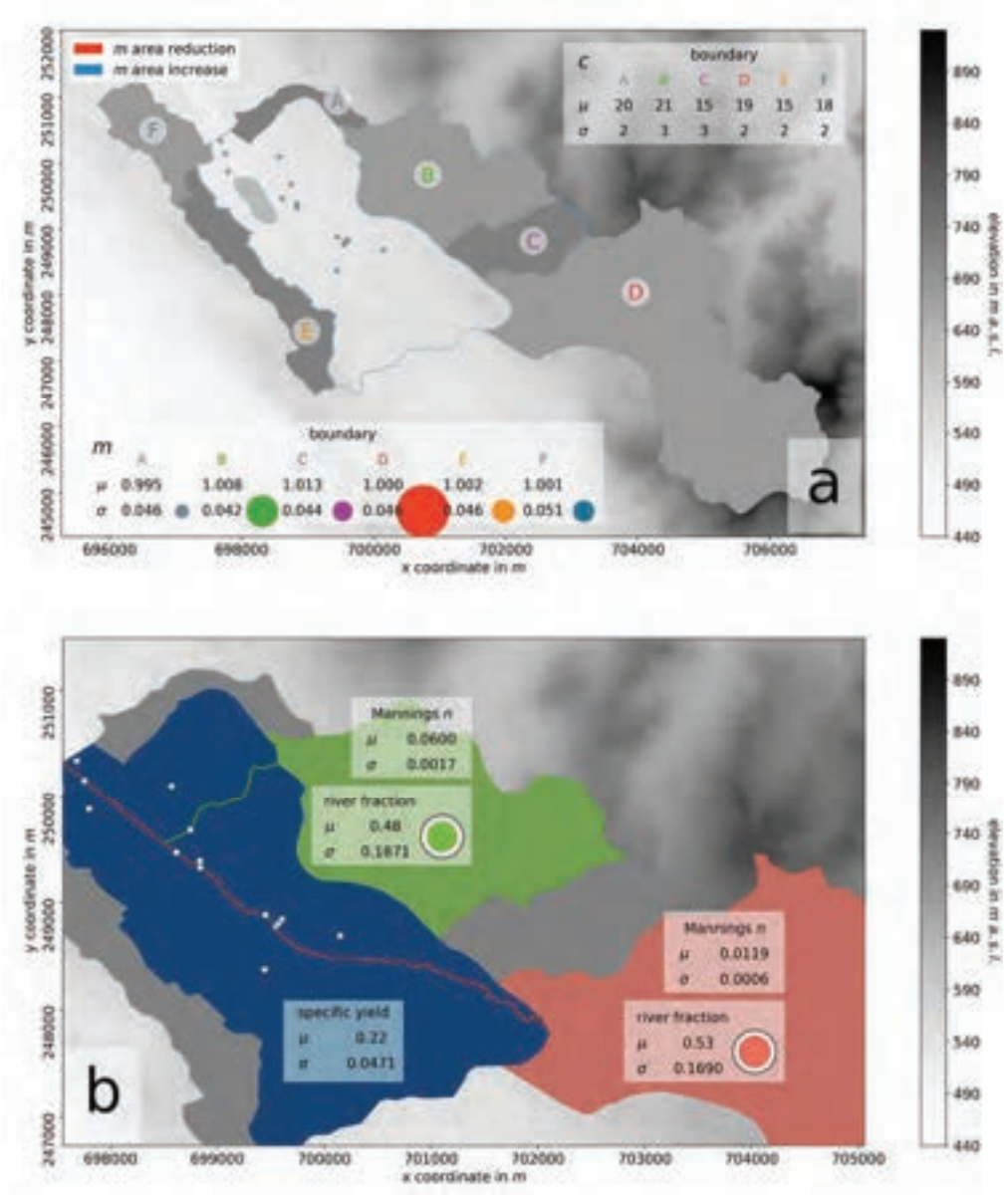


Figure III-8. Prior values for specific yield and the forcing-related parameters at the final iteration for $N = 100$. Subplot (a) illustrates the boundary-related parameters: the scalar multiplier m and the recharge delay factor d in hours; mean m is illustrated as a faint outline around the boundaries, visualizing the area inflation or deflation. Subplot (b) shows the specific yield and river-related parameters: the river discharge fraction and the Manning's number for Wildbach and Luppmen, respectively.

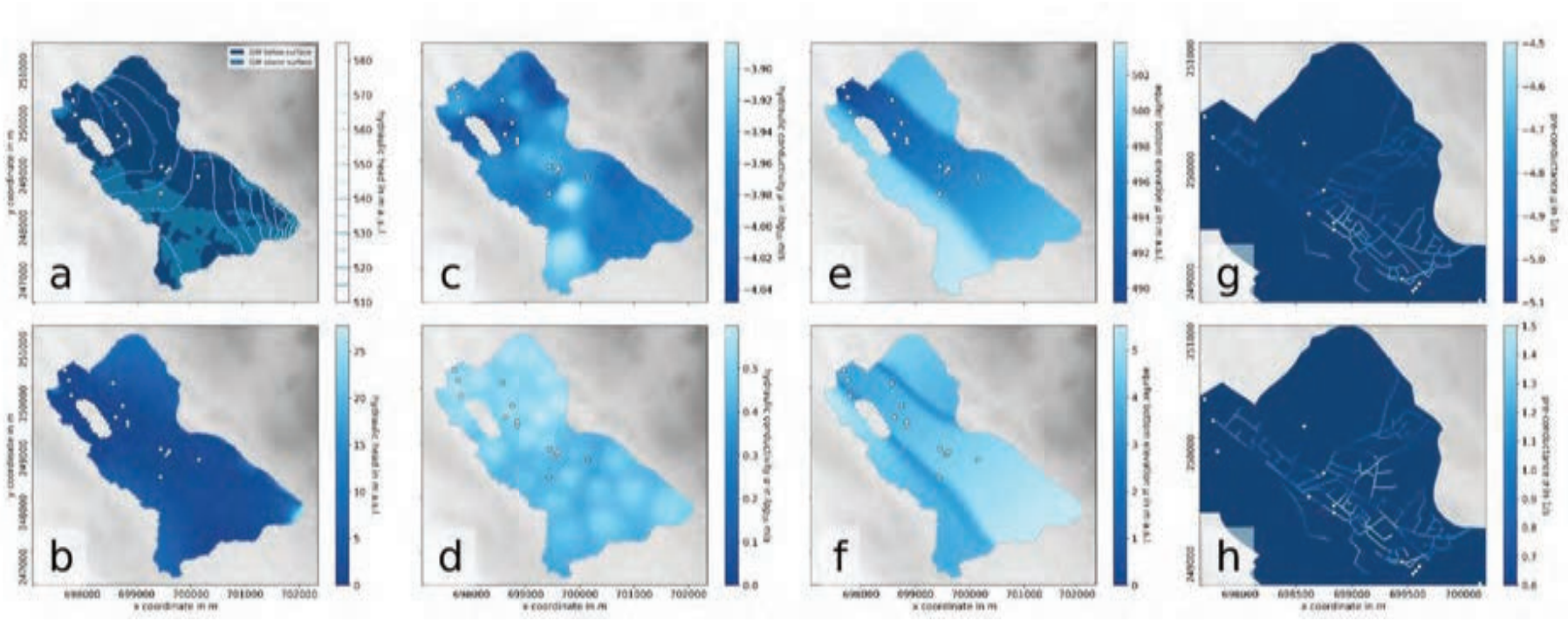


Figure III-9. Prior parameters and hydraulic head for $N = 30$. The two rows illustrate mean (a, c, e, g) and standard deviations (b, d, f, h) of hydraulic head in the initial steady-state simulation period (c, d), hydraulic conductivity (e, f), aquifer bottom elevation (g, h), and canalization conductance (i, j).

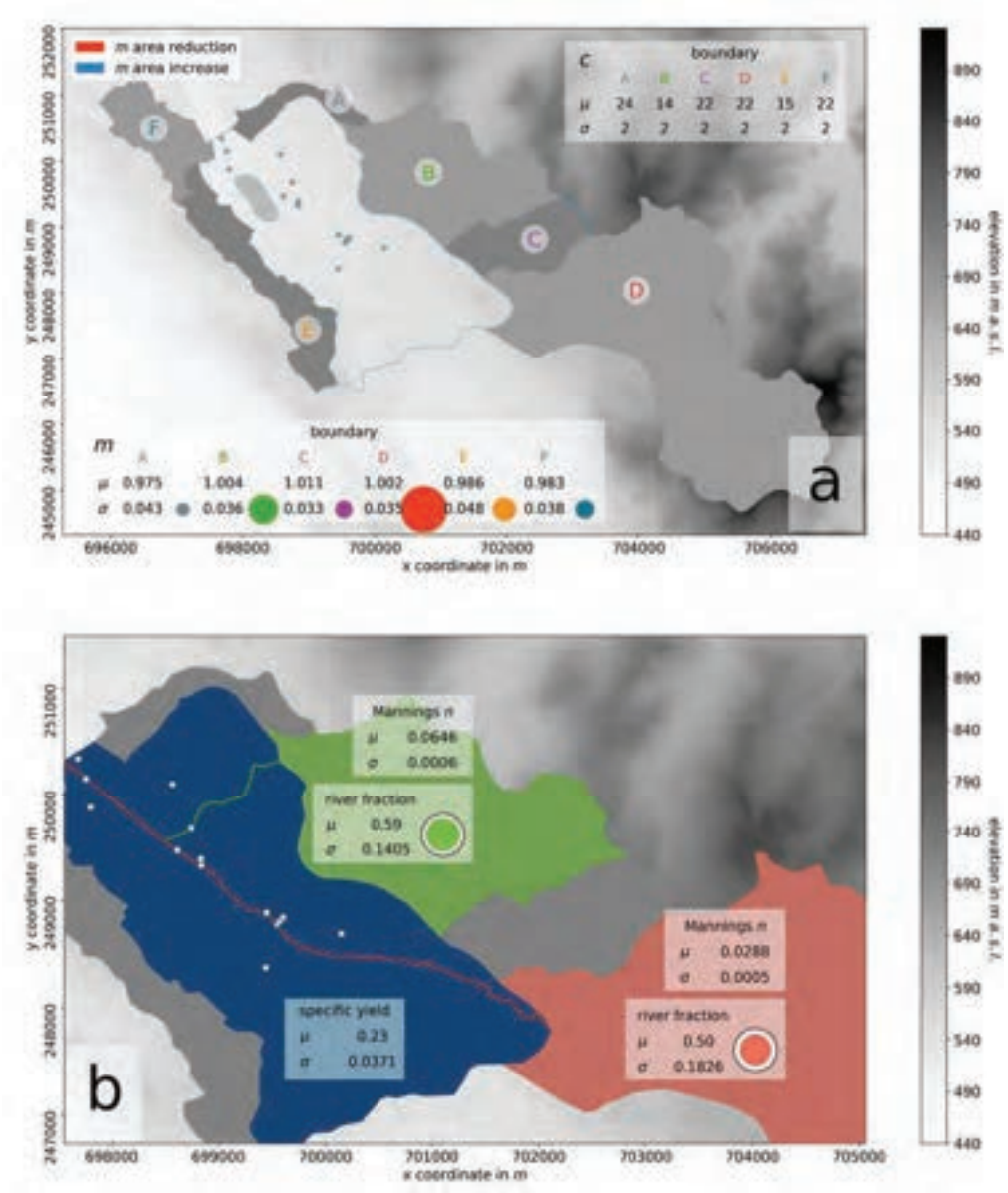


Figure III-10. Prior values for specific yield and the forcing-related parameters at the final iteration for $N = 30$. Subplot (a) illustrates the boundary-related parameters: the scalar multiplier m and the recharge delay factor d in hours; mean m is illustrated as a faint outline around the boundaries, visualizing the area inflation or deflation. Subplot (b) shows the specific yield and river-related parameters: the river discharge fraction and the Manning's number for Wildbach and Luppmen, respectively.

Figure III-11. Change of parameter uncertainty through the 100 iterations for the scenario $N = 100$.

Figure III-12. Change of parameter uncertainty through the 100 iterations for the scenario $N = 30$.